Caracterización del generador de número aleatorios del Kilobot o *Moneda*

Tomás Ayala y Romina D'Alessandro

Corrimos `moneda.c`, usando siempre la función `rand_hard()`. Dejamos fijo el tiempo mínimo de prendido/apagado del LED del Kilobot a `TIME = 1000` (milisegundos). Registramos varias filmaciones de 15 minutos del Kilobot "tirando la moneda" y una de 1 hora.

De cada medición obtenemos un `.csv` que en sus columnas tiene (i) la intensidad (en un entorno) del LED, con valores entre 0 y 255, y (ii) el instante del tiempo correspondiente. Graficamos la evolución temporal $I(t)$, su autocorrelación (ver la función `autocorrelación` de `aux.py`) y la frecuencia con las que ocurren las consecutividades[1] de longitud n. Por ejemplo, si la moneda en una tirada arrojó HTHTTTHHTHHHHHH, la consecutividad de longitud $n = 1$ se presentó 4 veces (H,T,H,T); la de $n = 2$, 1 vez (HH); la de $n = 3$, 1 vez (TTT); la de $n = 4$ y $n = 5$, ninguna y $n = 6$, 1 vez (HHHHHH). Notar que consecutividades de cara o ceca contribuyen por igual.

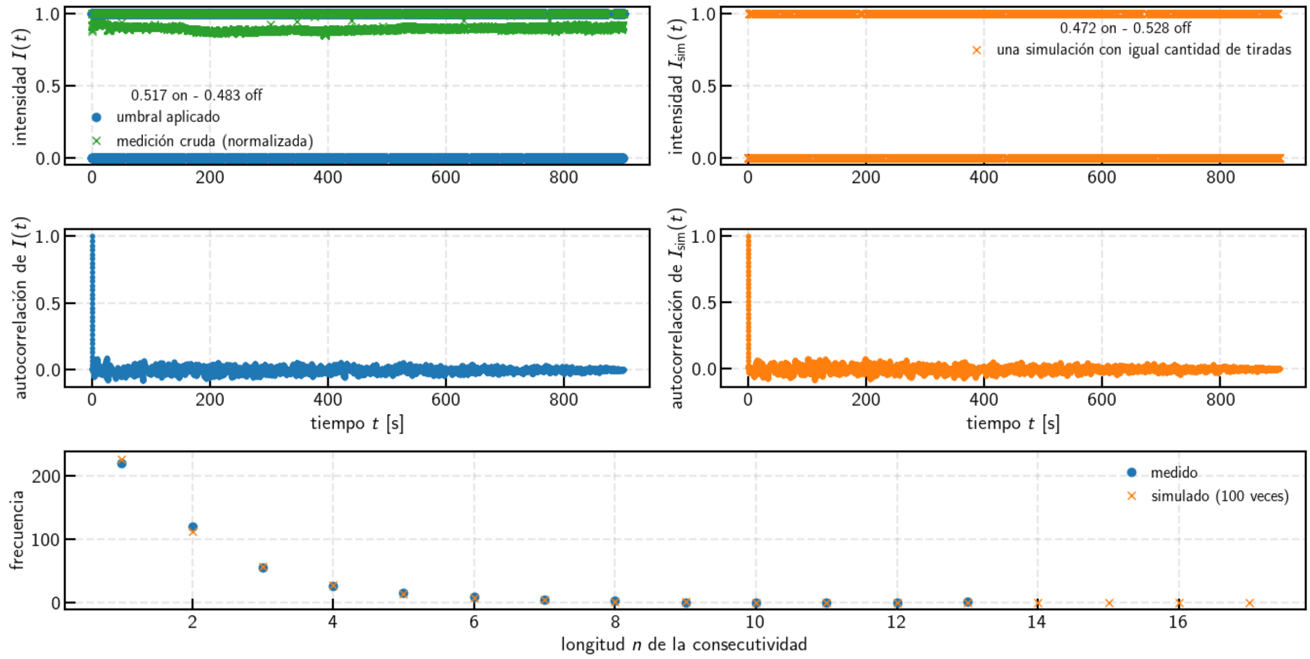

Figura 1. 15 minutos, medición b.

La medición cruda fue tomada, parece, con mucha luz ambiental. Por eso cuando el LED estaba apagado, el valor de la intensidad está más cerca de saturar[2] (255) que de cero. Aun así se logra distinguir los dos estados del Kilobot. Para el análisis le pasamos un umbral que asigna sólo dos valores: 1 (prendido) y 0 (apagado).

En naranja se presenta el resultado de una simulación de la moneda tirada 900 veces ya que 15 min = 900 s y era `TIME = 1000`.

Las autocorrelaciones, en ambos casos medido y simulado, caen a cero en 1 s o, equivalentemente, luego de 30 puntitos.[3].

En Labo 7 notábamos que ocurrían consecutividades en las tiradas de longitud $n = 12$ por ejemplo. Esto nos volvía locos, por eso es parte de la caracterización medir la frecuencia con las que estas consecutividades aparecen. También lo fue el cálculo de la autocorrelación: pensábamos que con altas consecutividades presentes, el tiempo de correlación iba a ser mayor al de una moneda "ideal" (i.e. simulada en python), pero no.

Nos focalizamos entonces en el conteo de consecutividades. Para esta medición, comparamos esta frecuencia de aparición de consecutividades con las de una moneda simulada. Presentamos, en naranja, la media de 100 realizaciones, cada una con la misma cantidad de tiradas[4]. Me sorprendió mucho que las frecuencias medidas y simuladas concuerden tan bien.

[1] ¿Existe esta palabra?

[2] Los valores presentados en verde están normalizados, o sea, divididos por 255.

[3] Grabamos a 30 fps y los frames los capturamos con una tasa $t = 1$, o sea, guardábamos 1 de cada t frames, quedándonos en este caso con todos los frames.

[4] Si la cantidad de tiradas es muuuy grande es posible, aunque poco probable, que aparezca una consecutividad de, por ejemplo,

Resumiendo, lo medido (y de lo que sospechábamos) y lo simulado presentan el mismo comportamiento:

- 51.7 % on - 48.3 % off vs. 47.2 % on - 52.8 % off
- mismo tiempo de correlación = 1 s
- misma distribución de consecutividades (!)

Para las otras mediciones, los resultados fueron:

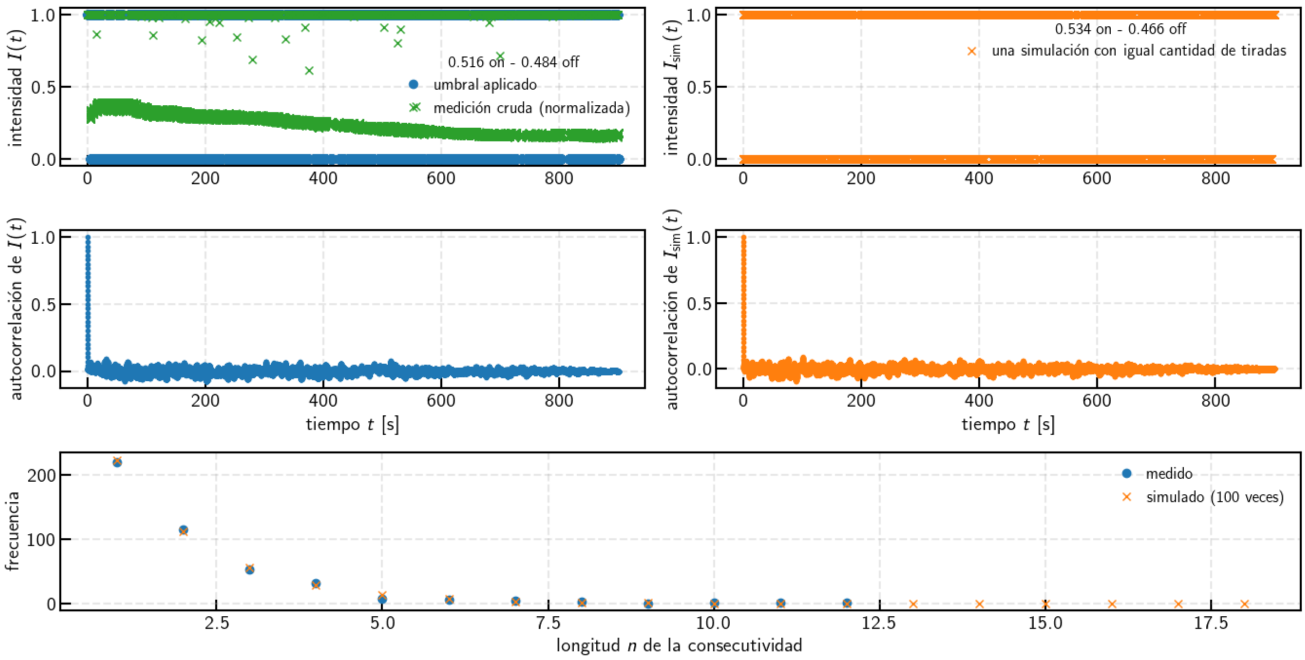

Figura 2. 15 minutos, medición c.

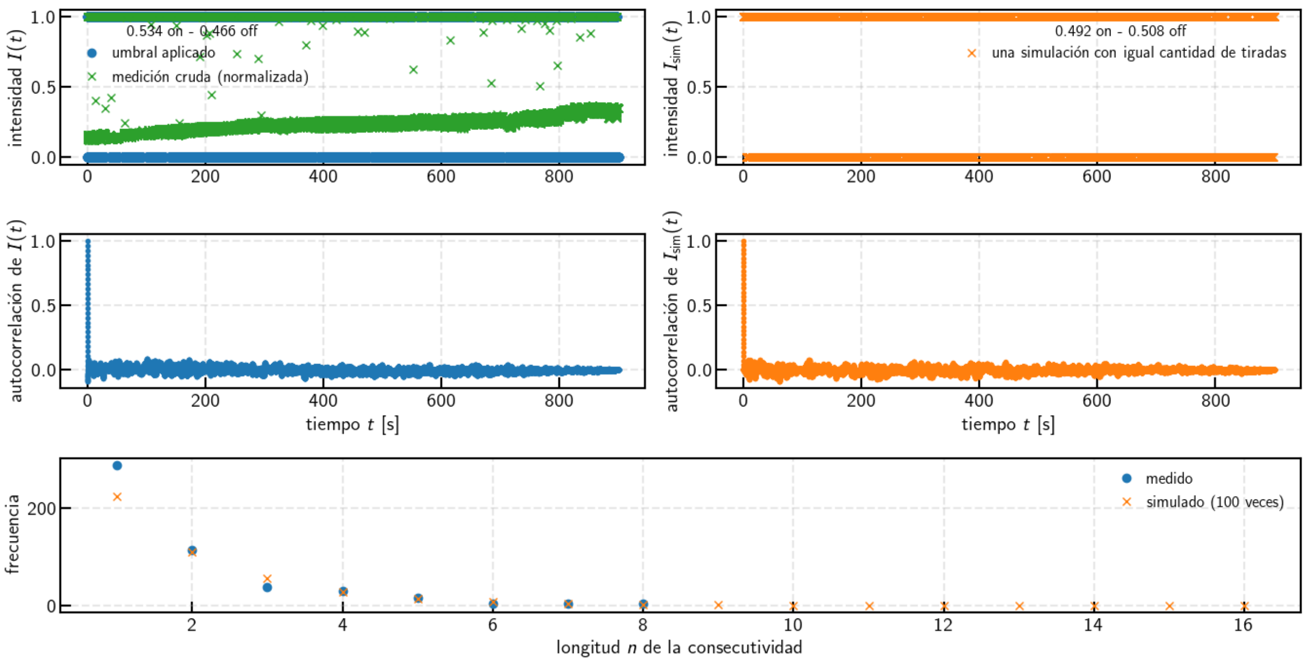

Figura 3. 15 minutos, medición d.

longitud $n = 20$, lo cual es casi imposible si la cantidad de tiradas es 20 e imposible si es menor. Por ejemplo, ocurrió por lo menos una vez una consecutividad de longitud 17 en un alguna de las 100 simulaciones.

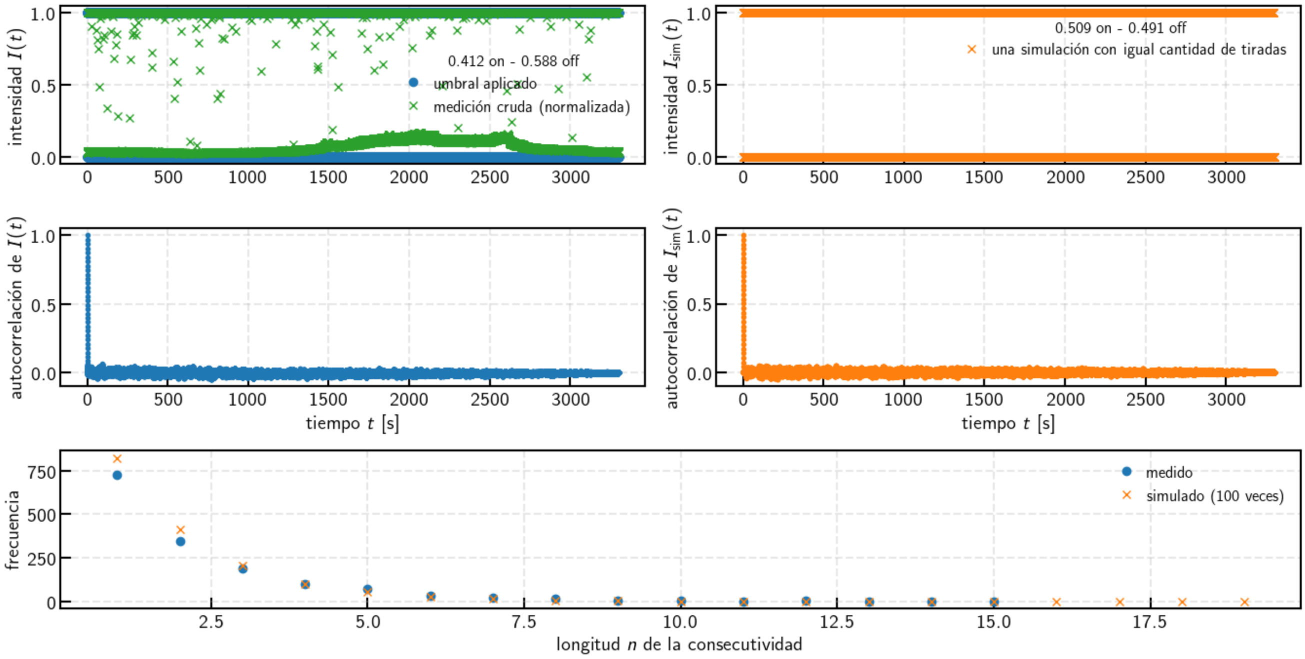

Figura 4. 1 hora.