# Static Guard

Supponiamo di avere un microfono che abbia come output alto/basso. Alto se il rumore supera una certa soglia, basso altrimenti.

Supponiamo di conoscere il tempo medio di attraversamento $T$, ossia il tempo che il transito di un'auto mantiene alto il microfono.

Sia $x$ il tempo di un'attivazione, ossia il tempo tra due valori bassi del microfono.

Si considerino i parametri $\alpha, \beta \in \mathbb{R}^+$

$\alpha$ è il parametro per gestire le attivazioni $x$ troppo brevi per essere considerati transiti, $\beta$ è il parametro che identifica il traffico.

Se $x < \frac{1}{\alpha}T$ allora l'attivazione è troppo breve per essere considerata un passaggio, quindi si ignora $x$

Se $\frac{1}{\alpha}T \leq x \leq \beta T$ allora $x$ è un singolo transito di un auto (invio al server un solo transito). $\beta$ è utilizzato per considerare la possibilità di auto con velocità minore della velocità media (ci mettono più tempo a transitare), mentre $\alpha$ è utilizzato per considerare auto con velocità maggiore della media (ci mettono meno tempo a transitare).

Se $x > \beta T$ allora **traffico**.

## Calcolo di x

In realtà il microfono che abbiamo preso non restituisce alto/basso, ma un range di valori da `0` a `1023`

In realtà, empiricamente a 5V restituisce valori tra `0` e `512` con media (empirica) di circa `250` (=baseline).

Quando c'è un rumore diverso dalla baseline ci sono oscillazioni nei valori.

$x$ può essere calcolato mantenendo un timer che si arresta dopo che per `K` misurazioni non rileva un valore diverso dalla baseline rispetto ad una soglia $\tau$

## Gestione del traffico

Quando si rientra nella casistica traffico bisogna valutare il numero di auto che sono passate dal tempo di percorrenza. Purtroppo senza conoscere la velocità di scorrimento del traffico direi sia impossibile.

**Soluzione semplice**: battezziamo una velocità di percorrenza del traffico (es. 15/20 km/h) e suddividiamo di conseguenza $x$ per ottenere il numero di auto.

**Soluzione pazza**: (vado a capo perché sarà una cosa lunga)

Visto che i valori non sono realmente alto/basso, idealmente si potrebbe provare a mantenere in memoria i valori di $x$ di una situazione di traffico. Bisognerebbe poi provare iterativamente a dividere in $k$ parti $x$ e confrontare tramite prodotto scalare (vedendo come vettori i campionamenti del suono fatti nelle $k$ parti) se questi sono simili. Nel caso lo siano, si è trovato un simil-pattern di traffico, interpretabile come una singola auto che lentamente transita nella colonna di auto, da utilizzare per contare il numero di auto nel tracciato di valori $x$.

Ovviamente non è scontato che questo sia possibile, in tal caso si potrebbe utilizzare come fallback la soluzione semplice.

## Calcolo di T

$T$ può essere calcolato come mediana degli ultimi $n$ valori di $x$ (mediana per evitare gli outliers degli alti tempi dati dalle situazioni di traffico).

## Calcolo di alpha e beta

Probabilmente li imposterei tra 2 e 4 così da individuare auto che vanno da 2 a 4 volte più veloci o più piano, però dipende anche dalla velocità media di percorrenza, quindi forse potrebbero essere messi in relazione con $T$

## Impostazione di Gain e AR

Mi viene da dire AR più veloce possibile per rilevare quante più variazioni nel suono, mentre il Gain dipende dalla strada e va capito empiricamente nel momento dell'installazione.

## Problematiche note

### Granularità di x

Se la granularità di $x$ è estremamente fine e c'è davvero molto traffico potrebbe andare in overflow. Considerando 4 ore di traffico e una granularità dei millisecondi si sta dentro un `unsigned int` (4 miliardi e qualcosa):

$$4 \cdot 60 \cdot 60 \cdot 1000 = 14,4000,000$$

Non so se i millisecondi siano sufficienti (dipende da velocità e sensibilità del microfono).

Per stare dentro un `unsigned int` la granularità al massimo è di 10 microsecondi (per 1 microsecondo servono 14 miliardi di valori con 4 ore di traffico).

## Tolleranza ai rumori

Se il tempo di transito medio $T$ è troppo breve, allora $\frac{1}{\alpha}T$ potrebbe essere talmente basso che rumori "normalmente lunghi" (es. gracchio di uccelli) vengono contati come transiti.

## Memoria nella soluzione pazza

Mantenere nella memoria di Arduino i valori campionati di $x$ è impensabile (troppo poco spazio a disposizione), è poca anche per mantenere solo i valori fuori baseline.