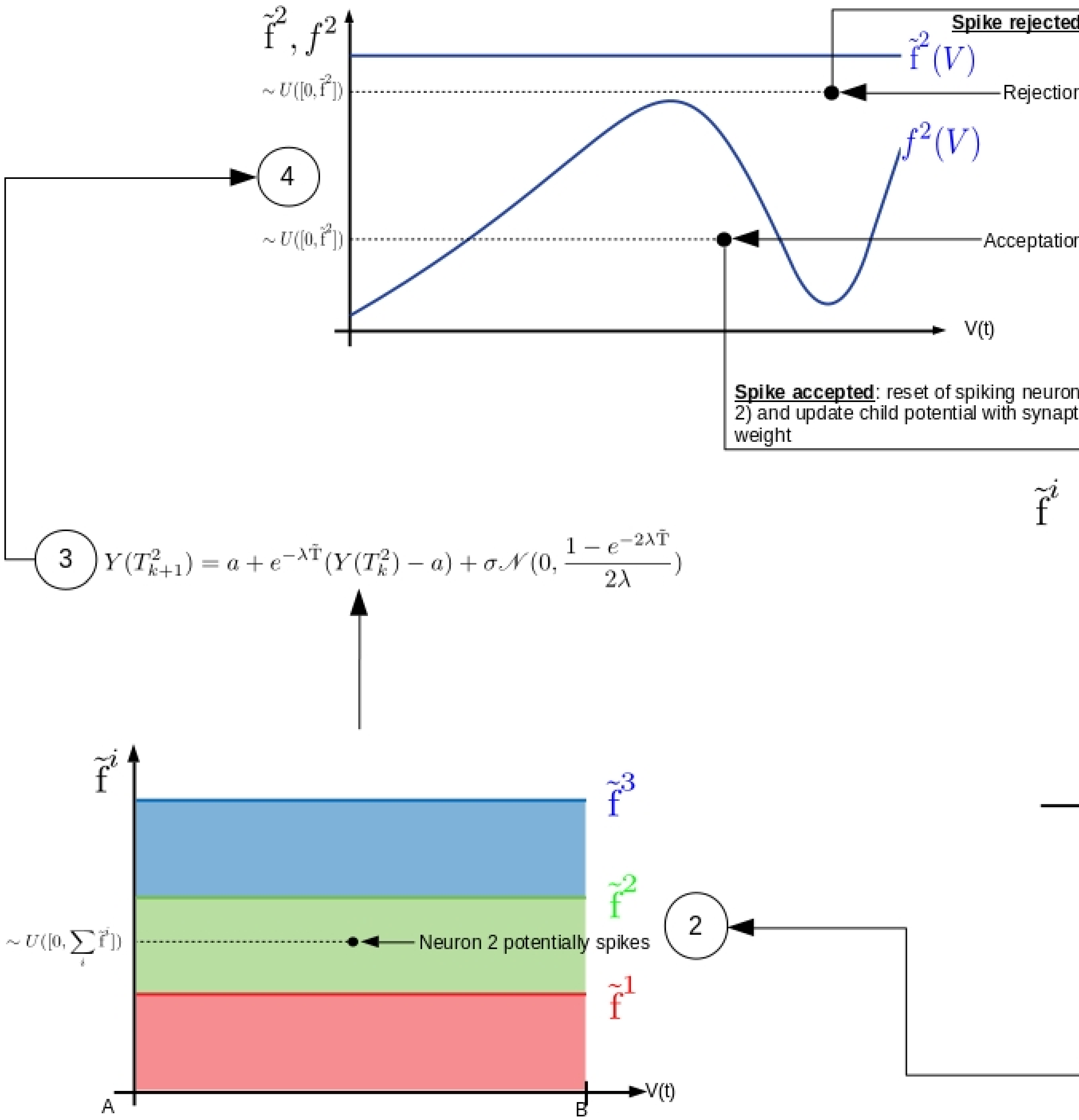
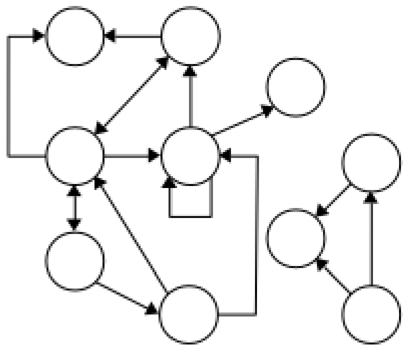
The algorithm is presented in figure **??**. It explicits the several steps for simulating the network of neurons. The first step (0) is looking for a time interval, where the next event shall be looked for. The separation between two sequential events follow a Poisson distribution of parameter the sum of approximation function $T_{k+1} = T_k + \tilde{\mathrm{T}}$, with $\mathscr{P}(\sum_i \tilde{\mathrm{f}}^i)$.

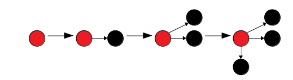
The complexity of clock-driven algorithms is of order N^2. Event-driven algorithms have much less complexity, of order N. But as our algorithm is based on athinning method, the complexity is higher, but still linear of N.



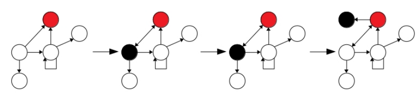
(a) Pseudo code of the thinning algorithm used for simulating the system

Initialize all parameters
repeat
  repeat ▷ Steps 0-1
    determine an interval $[t_{n-1}, t_n]$ on which sampling
    compute the array of $f_{max}(Y_{t_{n-1}})$ and $\sum f_{max}(Y_{t_{n-1}})$ on interval $[t_{n-1}, t_n]$
    $t_n \sim t_{n-1} + \mathscr{E}(\sum f_{max})$
  until not in good interval or all f_{max} are at 0
  $u \sim \mathbb{U}([0,1]) \rightarrow \mathbb{P}(\text{spiking neuron is neuron i}) = \frac{f^i_{max}(Y^i_{t_n})}{\sum_j^N f^j_{max}(Y^j_{t_n})}$
▷ Step 2
  $u \sim \mathbb{U}([0,1]) \rightarrow \mathbb{P}(\text{accepting spike of neuron i}) = \frac{f^i(Y^i_{t_n})}{f^i_{max}(Y^i_{t_n})}$
▷ Step 3

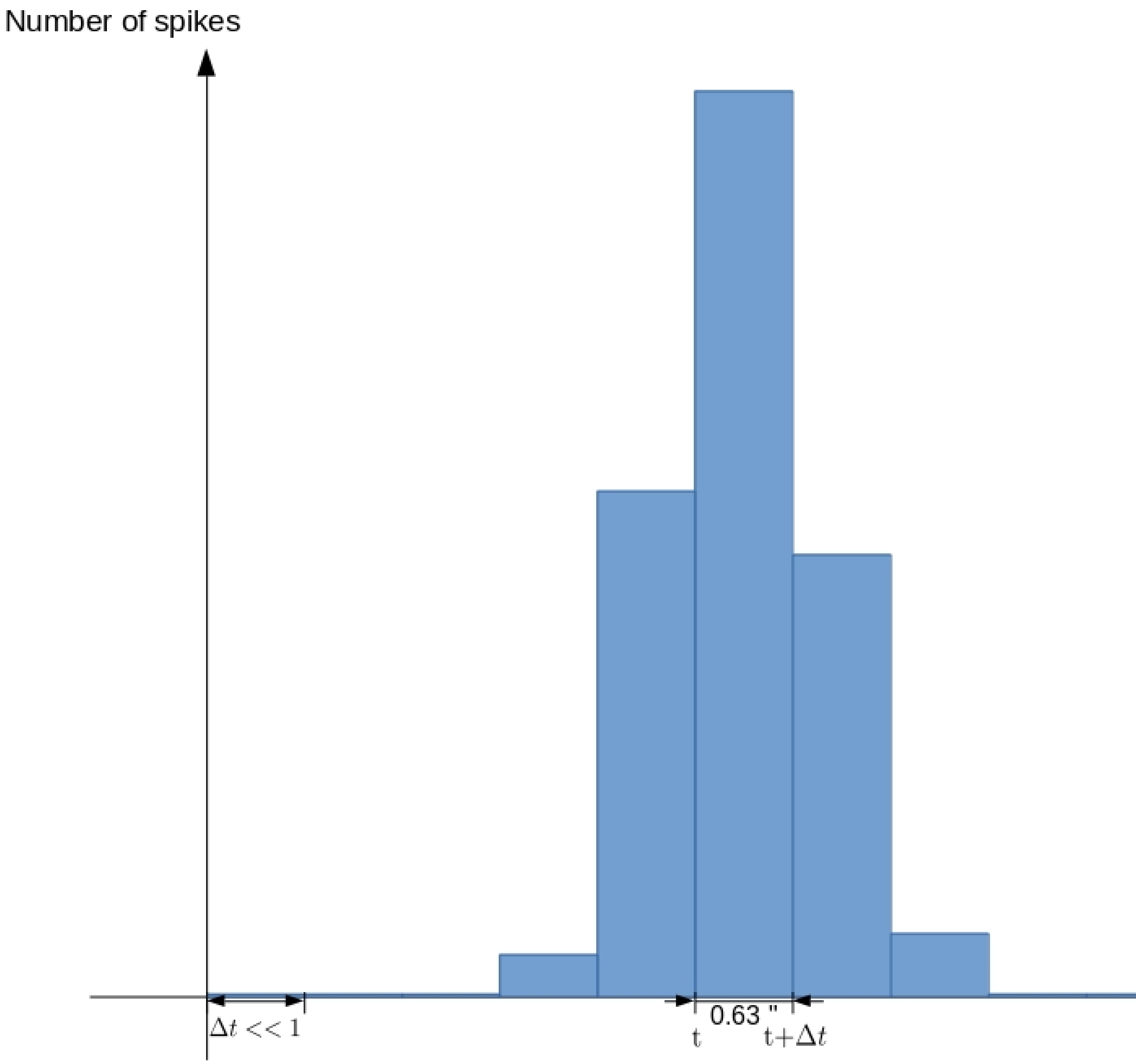




RNG(Status_0)
RNG(Status_1)
RNG(Status_2)
RNG(Status_{C_1})



RNG($\text{Status}_{C_1+C_2}$)
RNG($\text{Status}_{C_1+C_2+1}$)
RNG($\text{Status}_{C_1+C_2+2}$)
RNG($\text{Status}_{\sum_{k=1}^{3} C_k}$)



(a) Complexity as number of events during a time interval