

# Izhikevich Approximate Architecture

# Constants

- Limiting the implementation to only 4 patterns\* simplifies constant multiplication
- Constant multiplication can be realized using shifts and adds/ approximate adds
- A table for the constants, their approximations, and their representation as

\*Not all patterns have been tested, this architecture may be capable of producing all of them with some tweaks

Constant	Pattern A	PatternC	PatternE	PatternF
a	0.02	0.02	0.02	0.01
b	0.2	0.2	0.2	0.2
c	-65	-50	-55	-55
d	6	2	4	4
tau	0.25	0.25	0.25	0.25

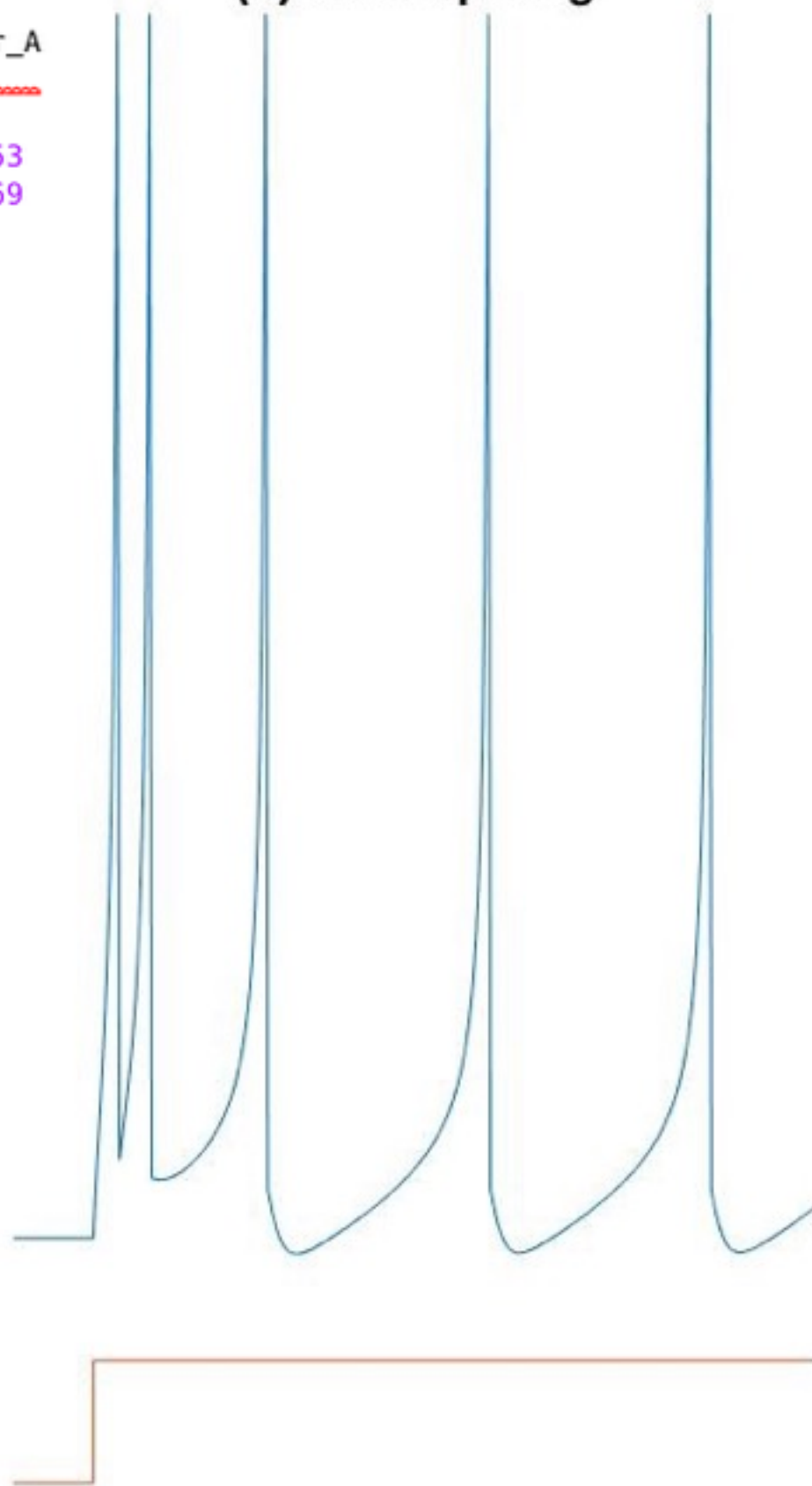
	Approximation	Representation
<b>a (0.02)</b>	0.0195	$2^{-6} + 2^{-8}$
<b>a (0.01)</b>	0.0117	$2^{-7} + 2^{-9}$
<b>b (0.2)</b>	0.2188	$2^{-2} - 2^{-5}$
<b>tau</b>	0.25	$2^{-2}$

(A) tonic spiking

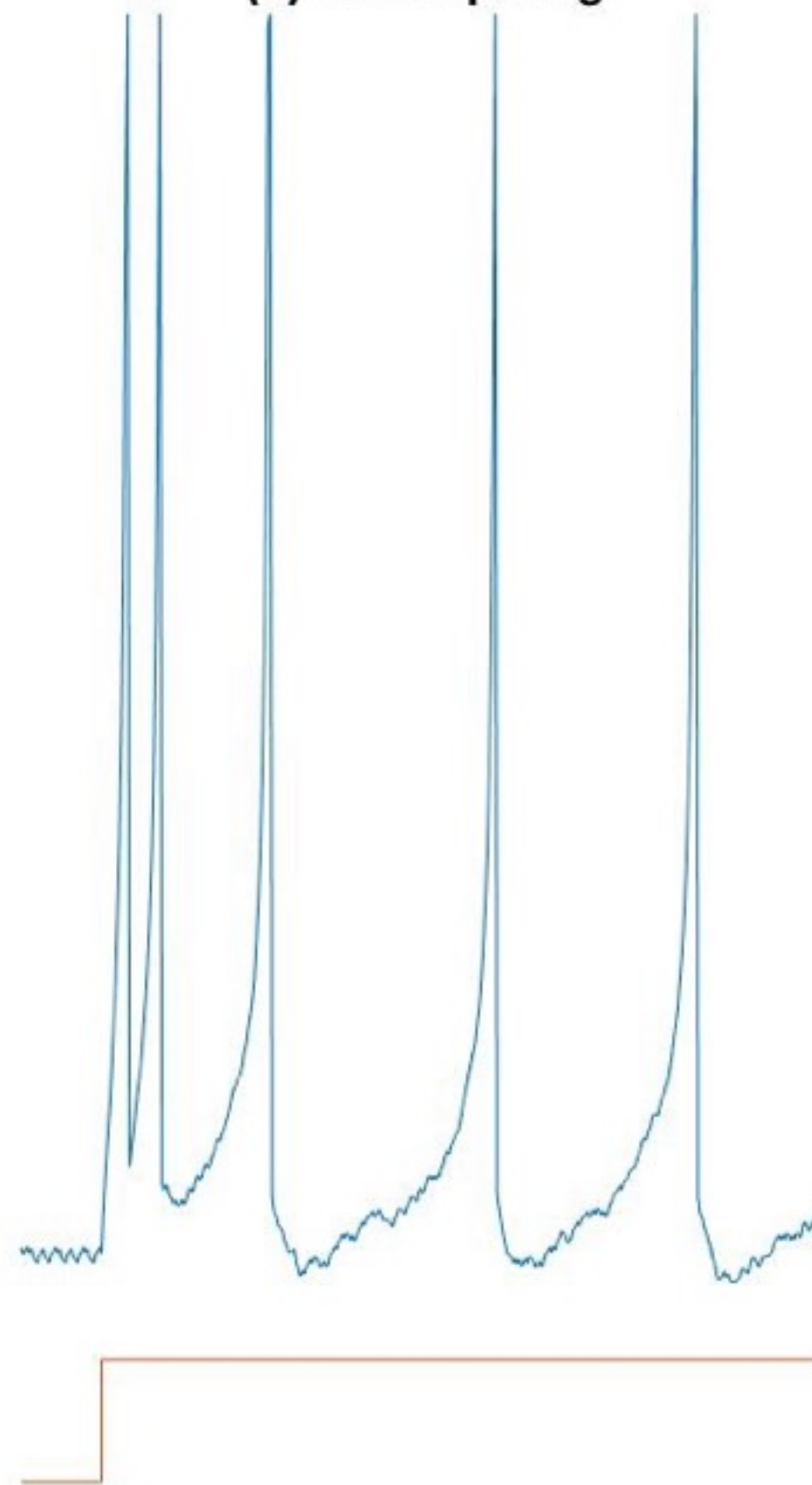
error\_A

xxxxxxxxxxxx

RSEE 3.2253  
MERRt 4.0869



(A) tonic spiking



**(C) tonic bursting EXACT**

error\_C

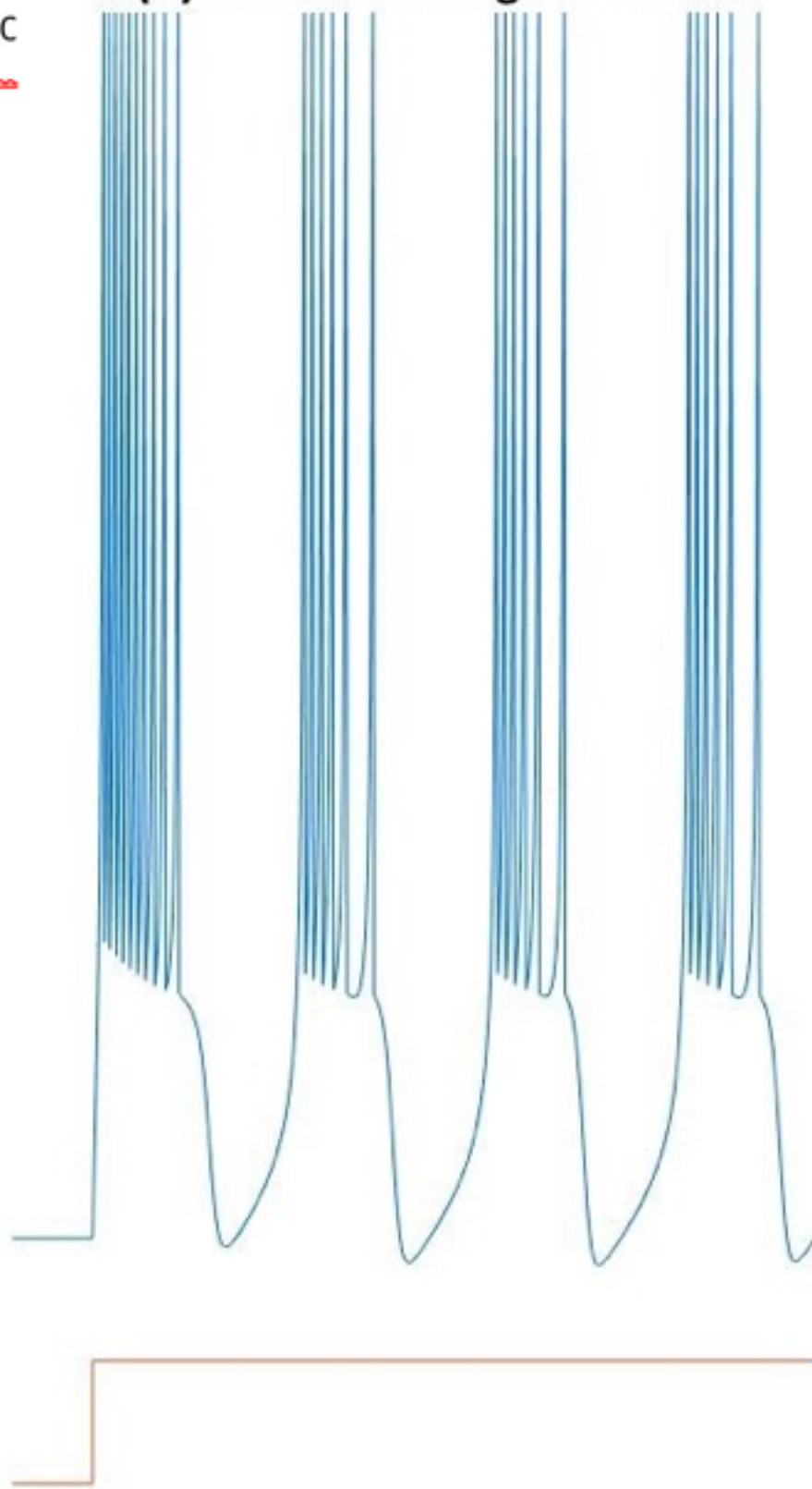
~~~~~

RSEE

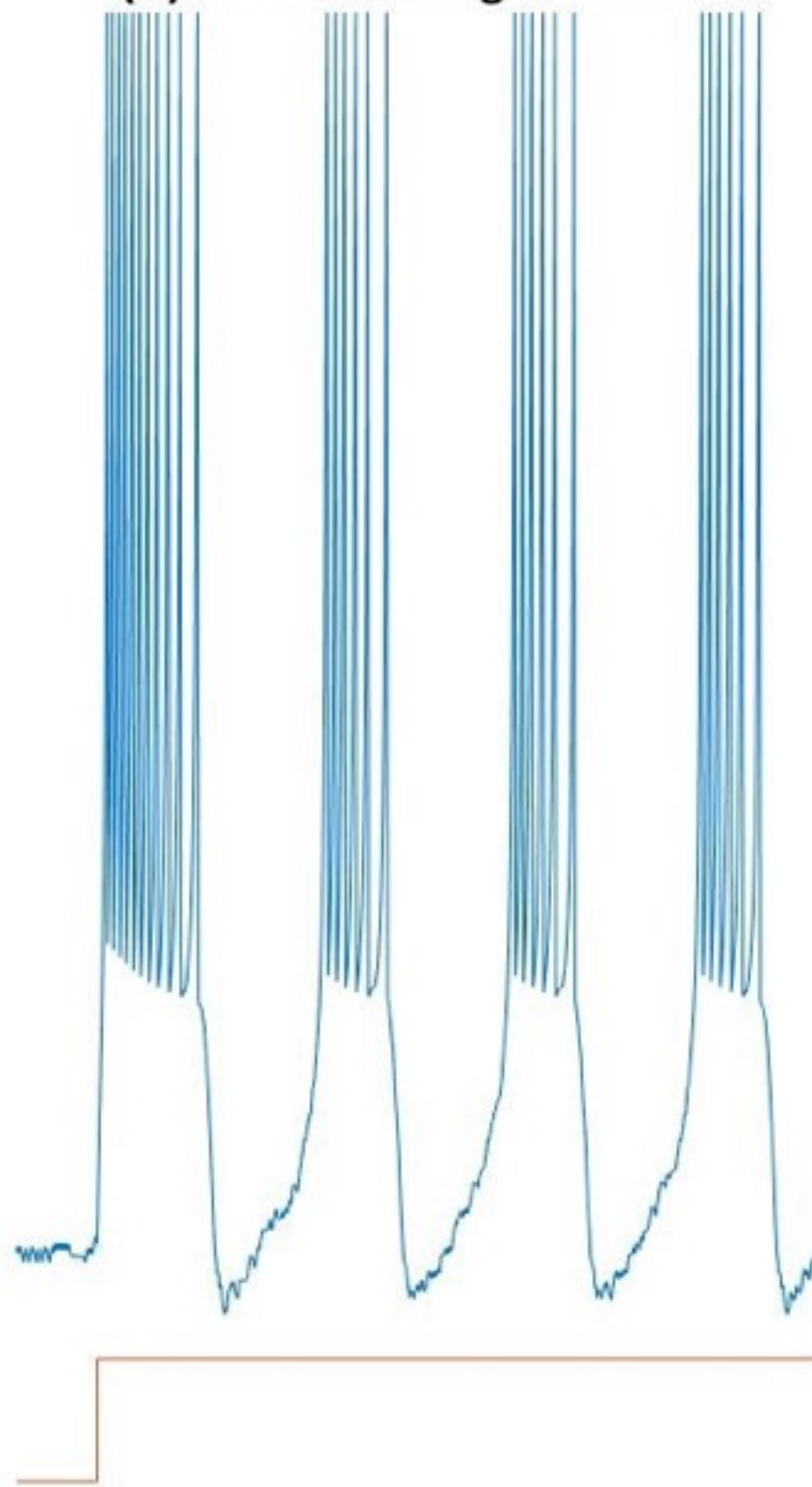
8.7381

MERRt

7.7832

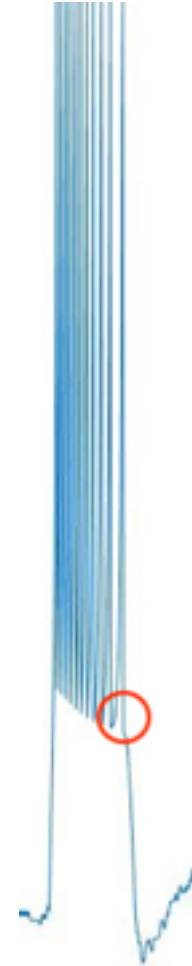


**(C) tonic bursting APPROX**



# Note on patternC

- There was an additional spike that was not take into account when calculating MERRt
- This is because that this spike does not correspond to another spike in the exact pattern



$$ERRt = \left| \frac{\Delta t_p - \Delta t_o}{\Delta t_o} \right| \times 100, \text{ where } \Delta t = t_{apex2} - t_{apex1} \quad (16)$$

$$MERRt = \frac{1}{n} \sum_i^n ERRt_i \quad (17)$$

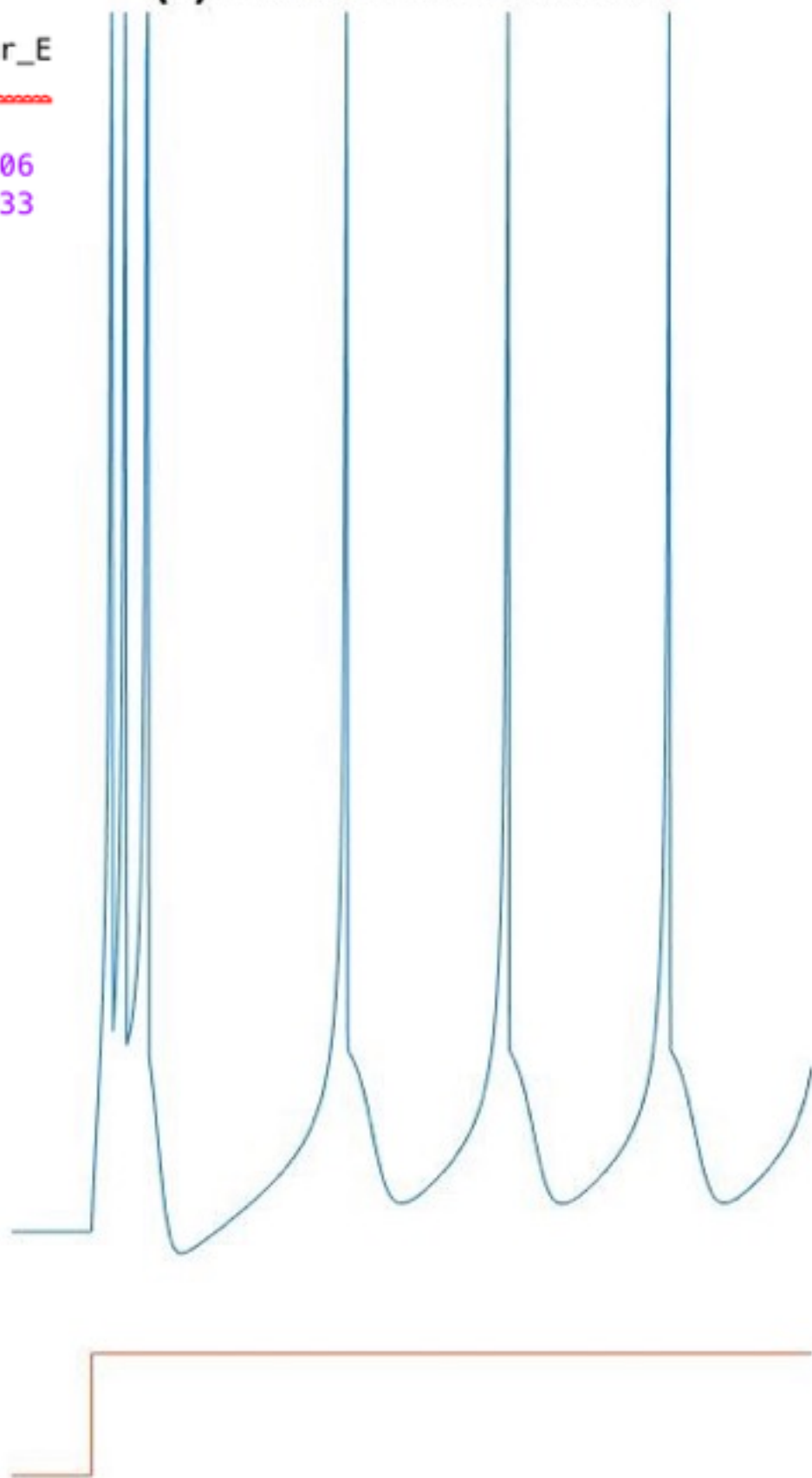
Where  $\Delta t_p$  and  $\Delta t_o$  represent the time intervals between two consecutive spikes in approximated and original models respectively.

(E) mixed mode EXACT

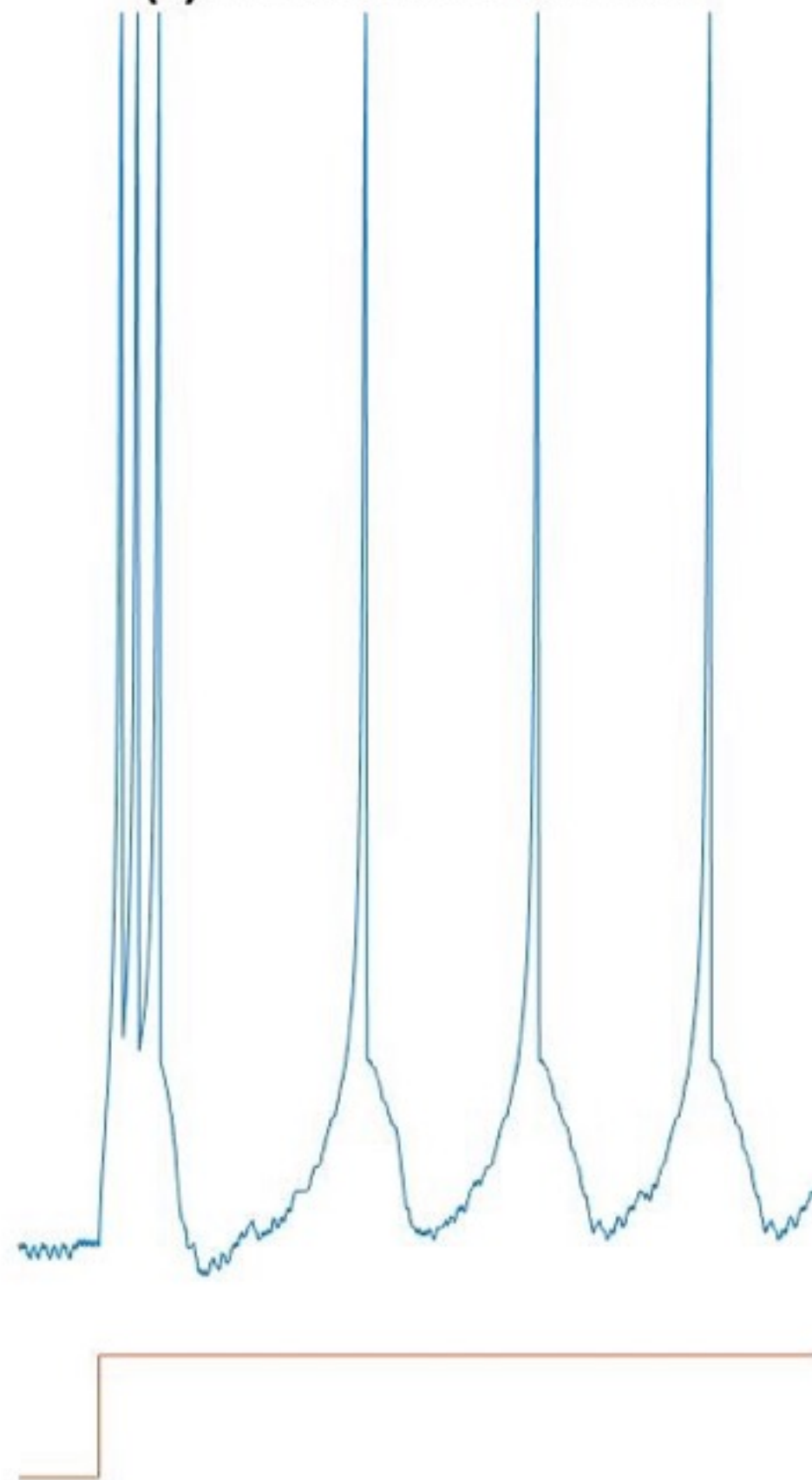
error\_E

-----

RSEE 10.06  
MERRt 9.0833



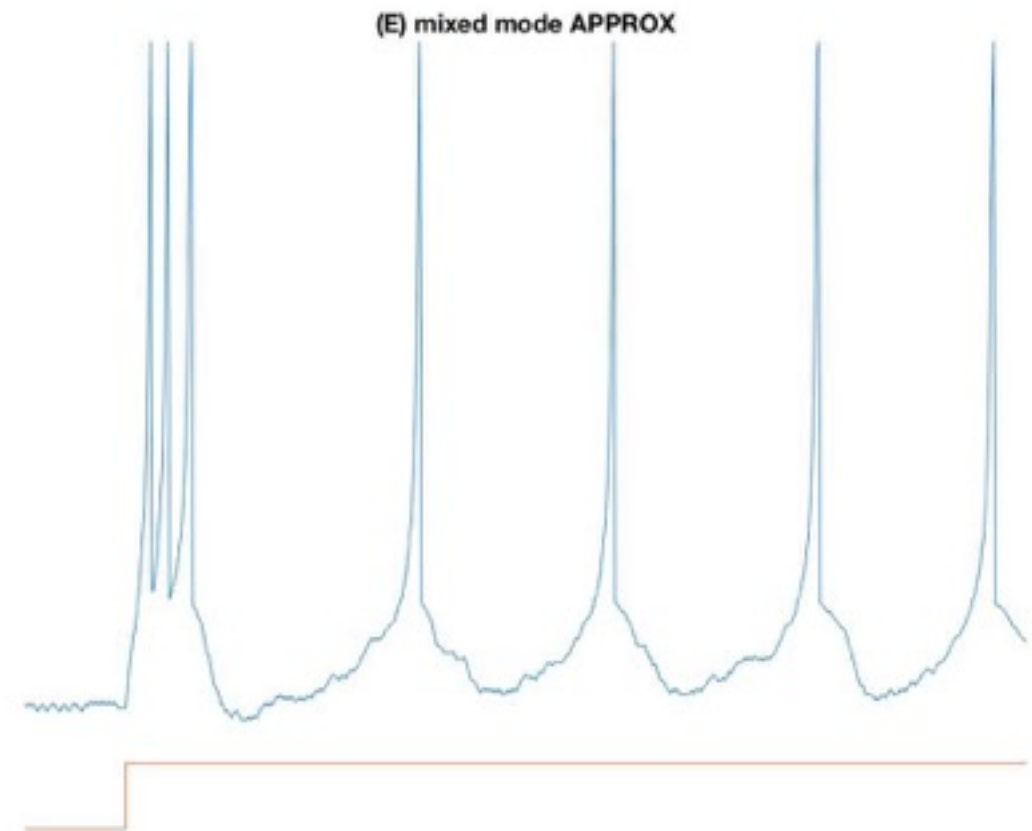
(E) mixed mode APPROX





# Note on pattern E

- Approximate pattern E is faster than the exact pattern
- To calculate MERRt and RSEE the last spike was omitted by limiting simulation time from 160 ms to 140 ms



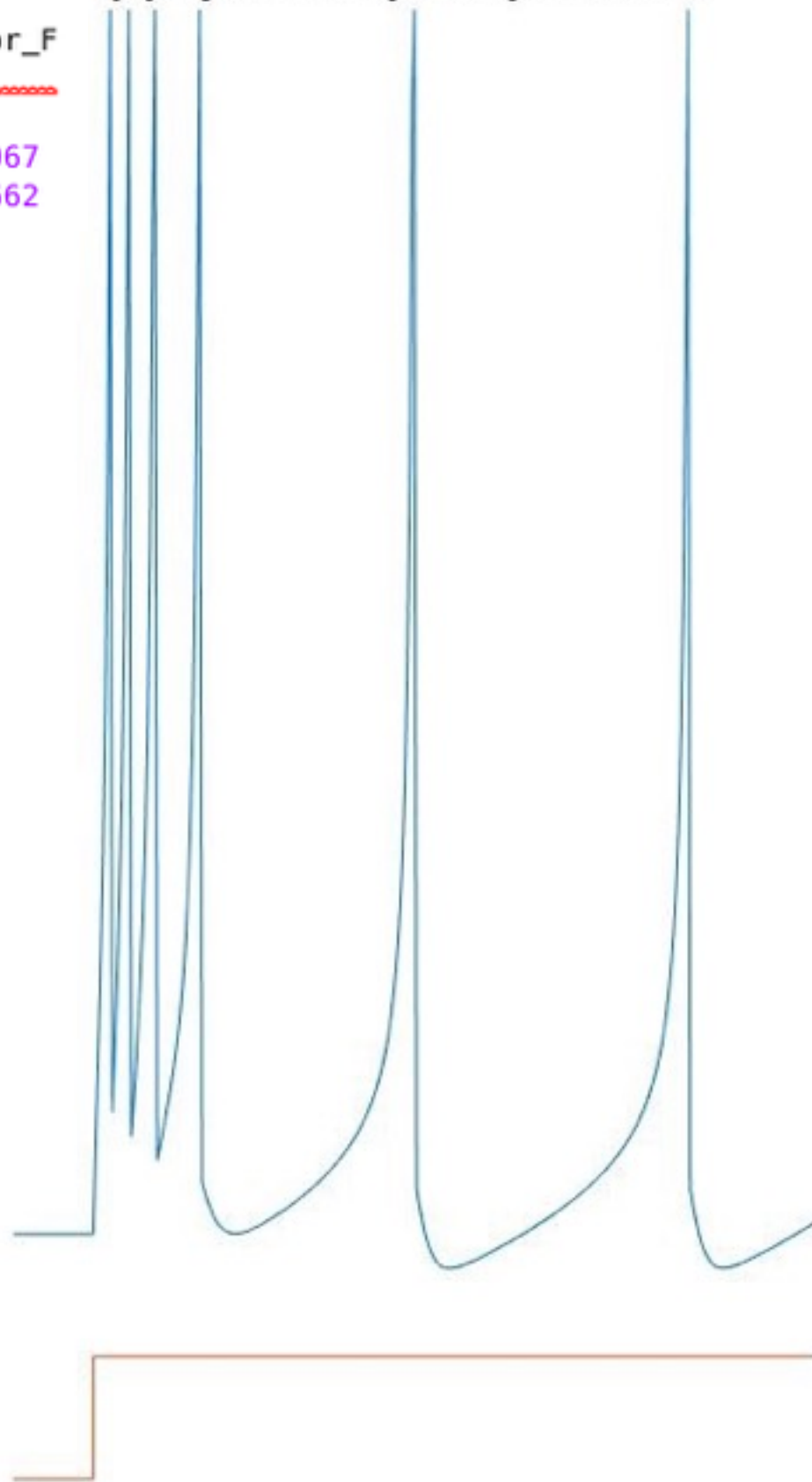
(F) spike freq. adapt EXACT

error\_F

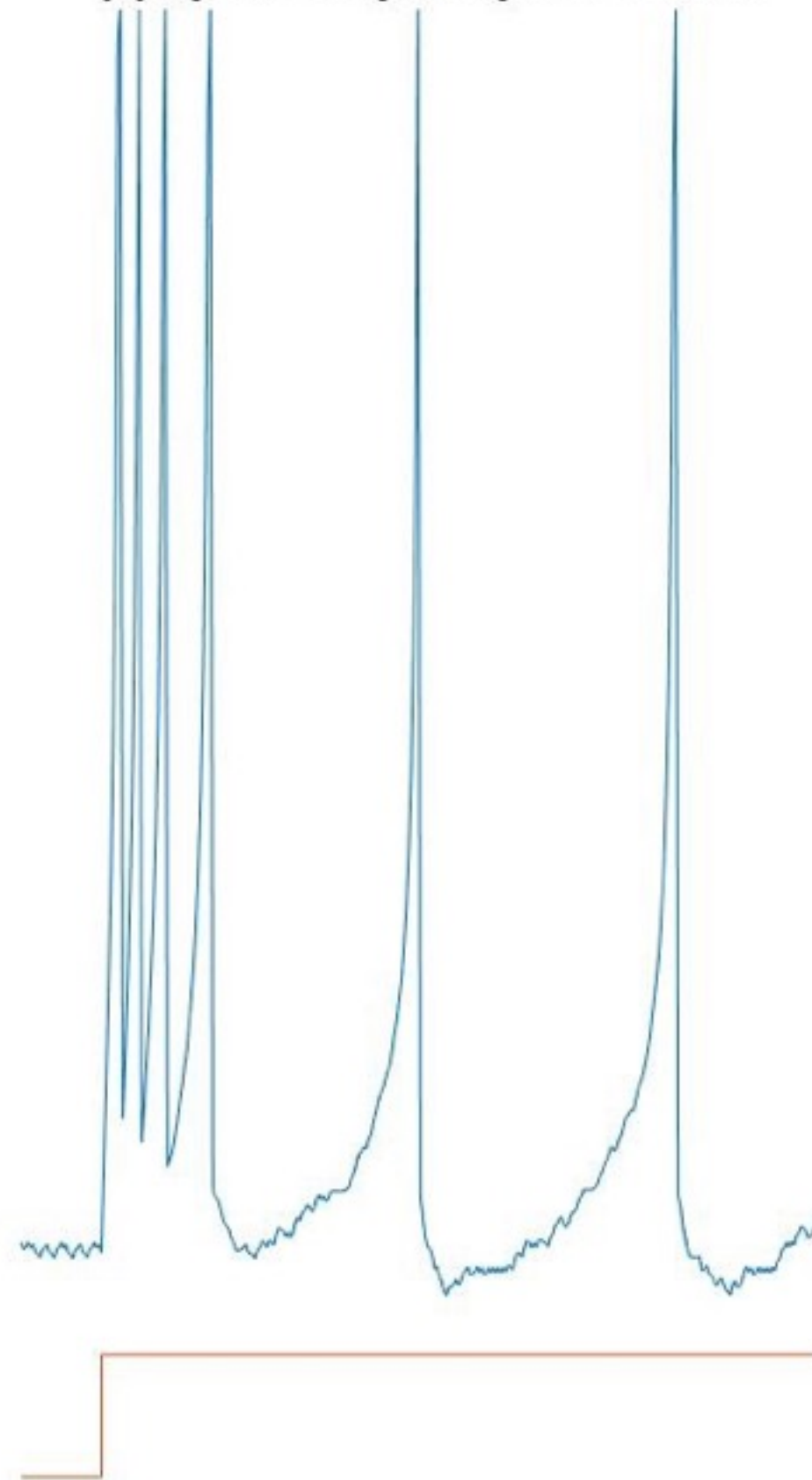
~~~~~

RSEE 3.6067

MERRt 1.9662

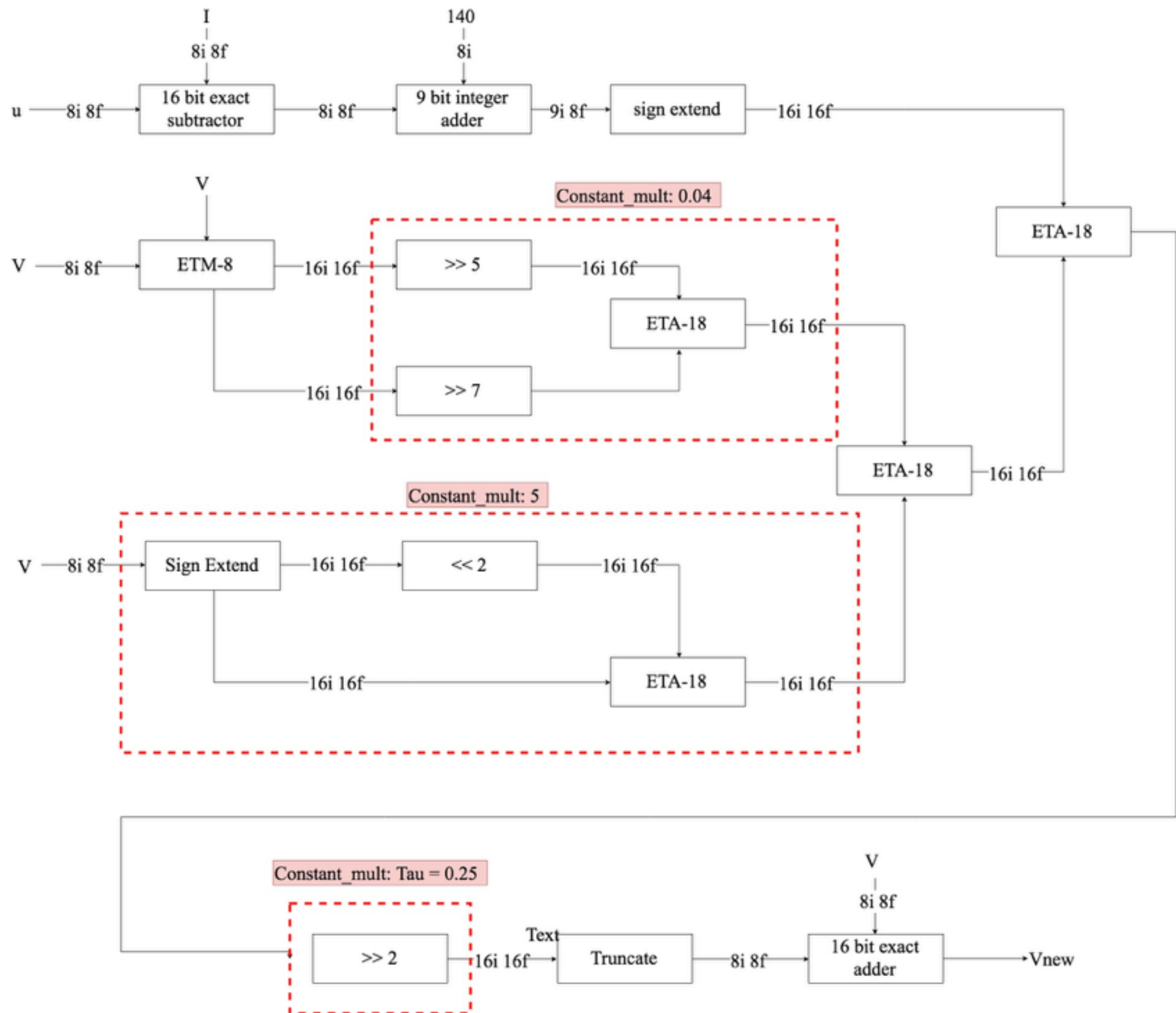


(F) spike freq. adapt APPROX



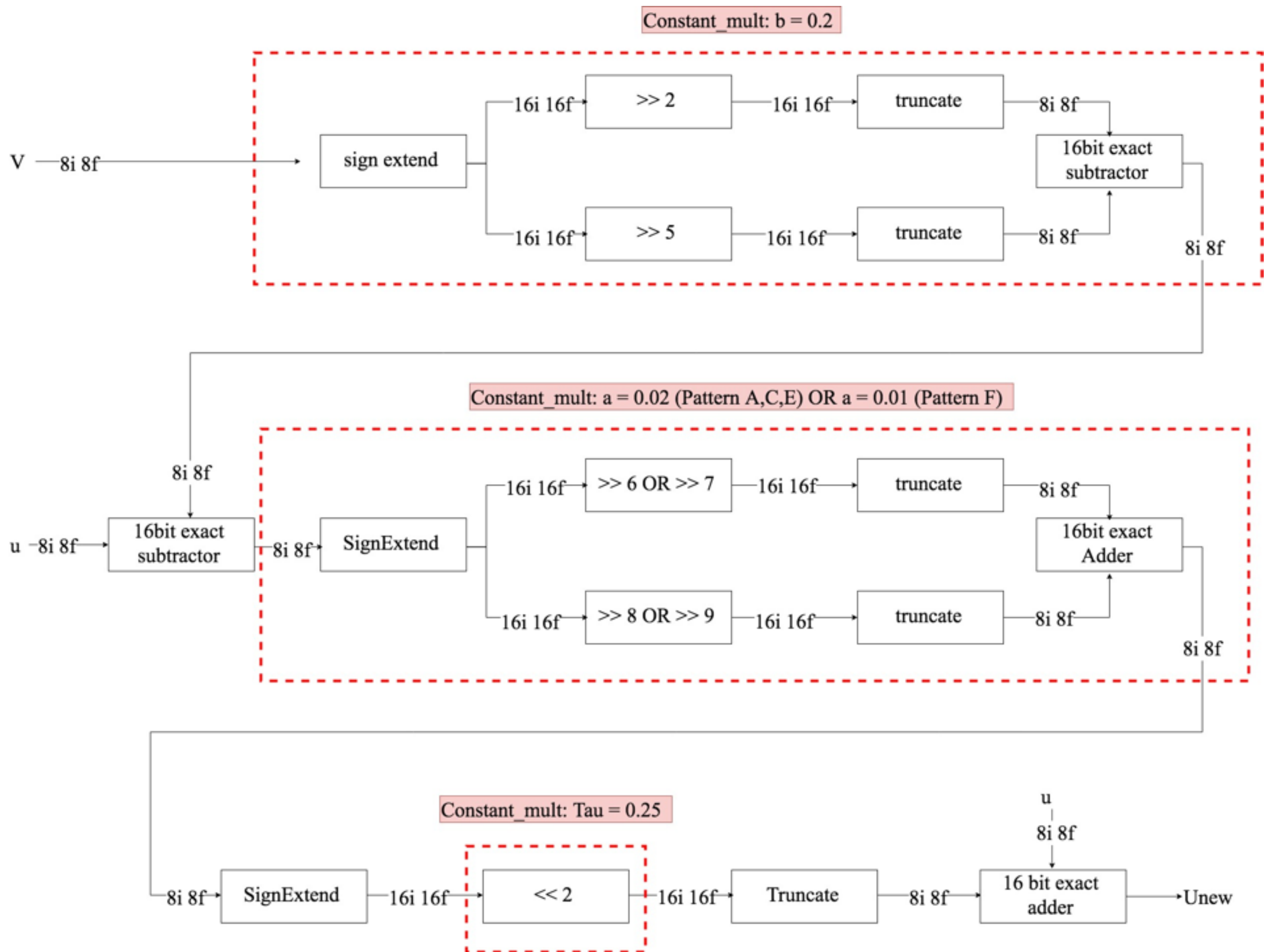
# V equation implementation

- 4X 18 bit adder
- 1X 16 bit subtractor
- 1X 16 bit adder
- 1X 9 bit adder
- 1X ETM-8 multiplier
- 1X 16bit approximation unit
- 4X 14 bit approximation unit



# u equation implementation

- 2 16 bit subtractors
- 2 16 bit adders



# Spike Detection Circuit

- 1 16 bit Adder
- 2 Muxes with 16bit bit width
- 2 16 bit registers
- 1 Constant Comparator (implementation to be realized using dc compiler)

