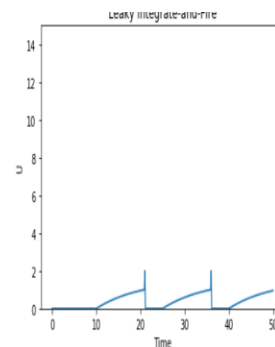


پیاده سازی مدل leaky integrate and fire:

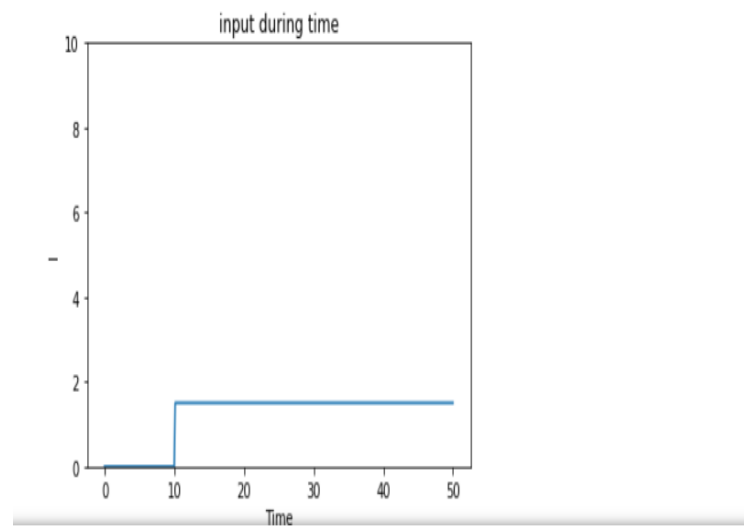
ابتدا پارامتر ها را تعریف میکنیم و سپس معادله ی دیفرانسیل IIF را به صورت زیر در میاوریم :

$$u[i] = u[i-1] + (-u[i-1] + I[i]*R) / \tau_m * dt$$

در صورت رسیدن به threshold اسپایک خواهیم داشت و به پتانسیل u برگشته و زمان را آپدیت میکنیم، این روند تا انتهای بازه ی زمانی ادامه خواهد داشت و برای 5 شدت جریان مختلف در حالی که سایر پارامتر ها ثابت بودند مقدار پتانسیل را اندازه گیری کرده و نمودار آن نسبت به زمان را رسم میکنیم (مشاهده میشود با افزایش جریان فاصله ی بین اسپایک ها کم میشود) برای نمونه خروجی یکی از نمودار ها برای اختلاف پتانسیل نسبت به زمان و جریان ورودی نسبت به زمان به صورت زیر خواهد بود:

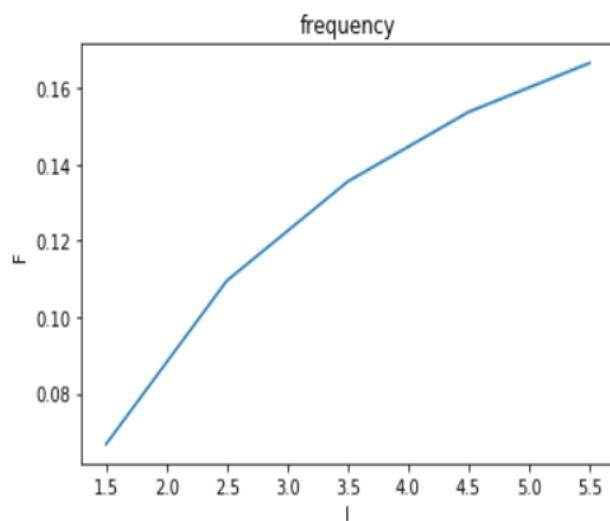


Time

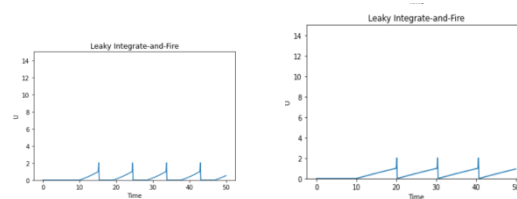


Time

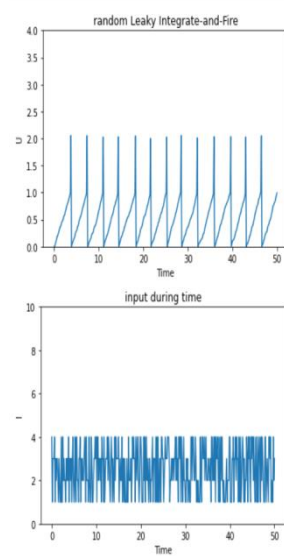
سپس باتوجه به زمان اسپایک ها مقدار فرکانس برای هر شدت جریان را حساب میکنیم $(1/(t_{spike1} - t_{spike0}))$ و سپس نمودار فرکانس به زمان را نیز رسم میکنیم که خروجی به شکل زیر خواهد بود:



در مرحله ی بعدی مقدار جریان ثابت خواهد بود اما سایر پارامتر ها مثل مقاومت و خازن هر بار تغییر میکنند و مشاهده میشود هر چه مقدارشان بیشتر شود فاصله ی بین اسپایک ها نیز بیشتر خواهد شد



در مرحله ی بعدی مقدار جریان را به صورت رندوم تغییر دادیم و نمودار پتانسیل به زمان و جریان به زمان را رسم کردیم:

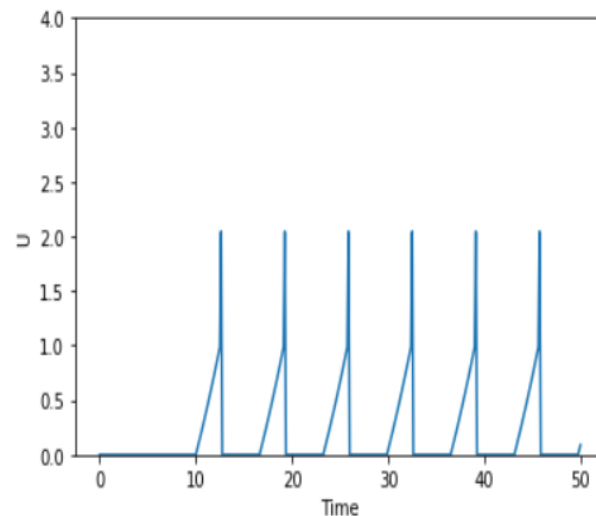


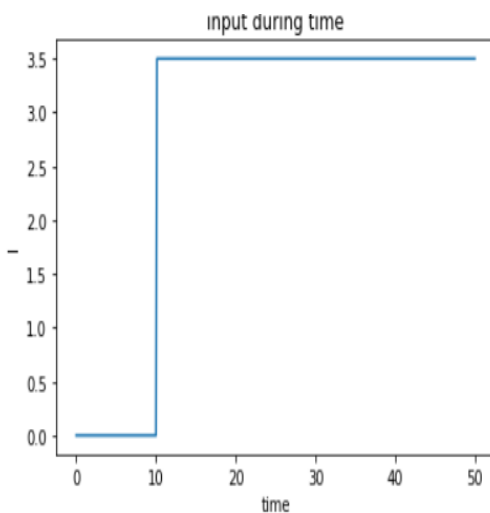
:Exponential integrate and fire

در این مدل دو threshold داریم اگر پتانسیل از پتانسیل rest بیشتر شود به شرطی که از threshold اول بیشتر باشد پتانسیل زیاد شده تا به threshold دوم برسد و اسپایک ایجاد شود که در آن صورت بعد از اسپایک t پدیت شده و همچنین به پتانسیل rest بازمیگردیم ، اما در صورتی که پتانسیل threshold اول کمتر باشد به سمت پتانسیل rest رفته و پتانسیل کاهش میابد، به دلیل سخت بودن حل معادله ی دیفرانسیل برای این حالت ، از تعریف مشتق استفاده کرده و با روش های iterative هر مقدار را با توجه به مقدار قبلی بدست میاوریم بنابراین معادله دیفرانسیل به شکل زیر در میاید و پارامتر های جدیدی نسبت به حالت پیشین نیز خواهیم داشت:

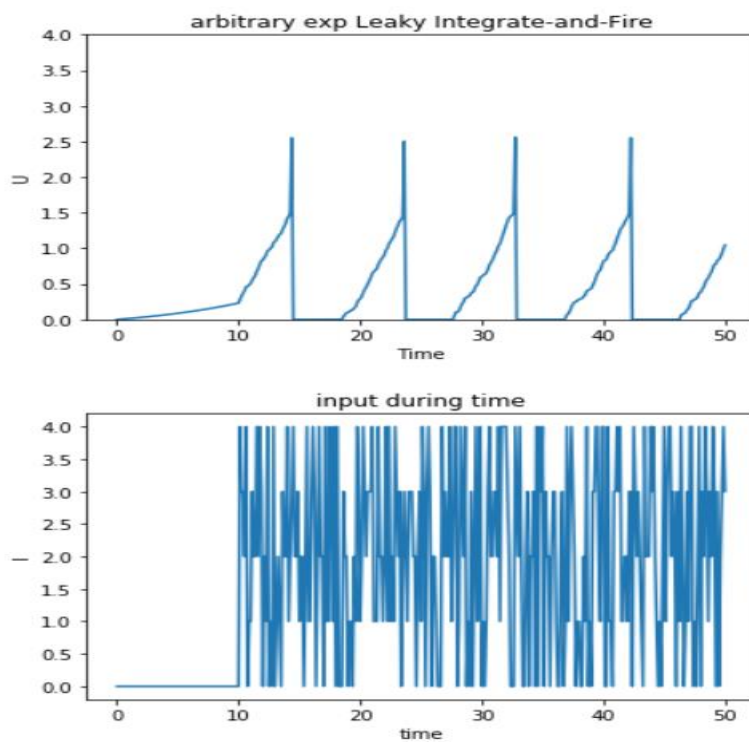
$$u[i] = (-(u[i] - u[i-1]) * dt / tm) + (R * dt * I[i] / tm + u[i-1]) + (dt * \Delta T * (\exp(u[i] - \text{firing_threshold}) / \Delta T)) / tm$$

در این حالت اگر ΔT به صفر نزدیک شود مدل مانند حالت پیشین عمل میکند ، همچنین در این مرحله نمودار پتانسیل به زمان و جریان به زمان برای شدت جریان های مختلف رسم کردیم (نمونه ی خروجی):

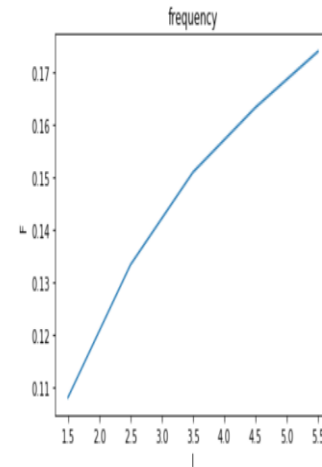




در مرحله ی بعد برای جریان های تصادفی مقدار پتانسیل را اندازه گیری کرده و نمودار آنها را رسم کردیم :



سپس نمودار فرکانس به زمان رسم شده است :



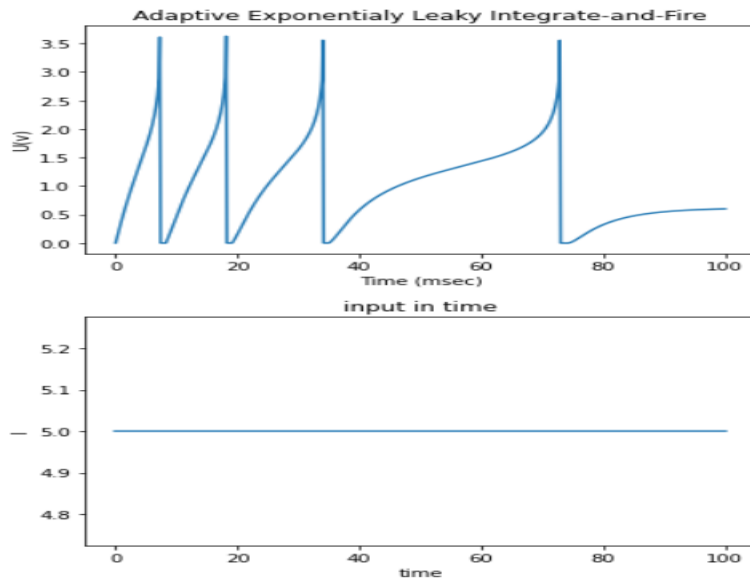
در مدل adaptive بعد از زدن هر اسپایک مقدار $R \cdot W$ از پتانسیل کم میشود بنابراین به دو معادله (یکی پتانسیل و دیگری مقدار W) نیاز داریم که در اینجا نیز بدلیل سخت بودن معادلات از روش های iterative استفاده میکنیم ، همچنین بعد از هر اسپایک به پتانسیل $rest$ برگشته و زمان را اپدیت میکنیم و counter همان مجموع تابع دلتاست (در صورتی که اسپایک داشته باشیم یکی به آن اضافه میشود):

$$w = w + (dt / tw) * (a * (u[i-1] - V_{rest}) - w + b * tw * counter($$

$$u[i] = u[i - 1] + ((-u[i - 1] + V_{rest} - R * w + I[i] * R + \delta T * (np.exp((u[i - 1] - \theta_{RH})/\delta T))) / \tau_m) * dt$$

در این مدل چون فواصل اسپایک ها برابر نبوده است نمдар فرکانس به زمان رسم نشده است همچنین در این مدل نیز نمودار پتانسیل به زمان با اعمال جریان های رندوم نیز رسم شده است

نمونه ی خروجی برای حالت جریان ثابت:



نمونه ی خروجی برای جریان رندوم:

