Osnabrück University

Neurodynamics

Course Code: 8.3047

Implementing the MAT-Model

Authors:	*Student IDs:*
Christian Johnson	981012
Jannik Schmitt	964461
Matthias Seidel	987174

August 18, 2020

Contents

1 Introduction

One of the biggest goals in the field of Neuroinformatics is the creation of models which mimic neuronal networks at a scale and speed comparable to that of the human brain. Accurate neuron models which avoid the obstacle of computation costs are required to satisfy the aforementioned goal.

The Multi-Timescale Adaptive Threshold (MAT) Model (Kobayashi et al., 2009) might be such a model, especially since the model can be adapted to a variety of different spiking behaviour including regular spiking, intrinsic bursting, fast spiking, and chattering. Based on the described attractive properties of the MAT-Model, an analysis of the model is desired to ensure a more detailed understanding of its functionality. Such analyses are targeted for different neurons and their spike responses to conclude on the predictive accuracy of the model.

In this report, we implement and evaluate the Multi-Timescale Adaptive Threshold Model in order to examine its functionality and accuracy in predicting spikes.

2 Problem Description

The overall problem description considered for this project is the implementation of the Multi-Timescale Adaptive Threshold Model and the subsequent evaluation of its predictions regarding the spiking behaviour of different neurons. The overall approach of the model implementation and its evaluation are variables which need to be determined upfront. The main tasks and goals of this project will be described in the following subsections.

2.1 Implementing the MAT-Model

The technical implementation of the MAT-Model is one the two main objectives of this project, which is also used as a baseline for all further analysis. The implemented model should provide as output the spiking behaviour of a neuron given a defined input current, where the neuron is described based on defined input parameters. The model should be able to represent a variety of different neurons (incl. Regular Spiking, Fast Spiking, Intrinsic Bursting and Chattering neurons) based on the given parameters, which ensures a wider range of analysis.

The MAT-Model consists of a non-resetting leaky integrator and a multiple-timescale adaptive threshold. The leaky integrator computes the voltage response to an injected input current and the adaptive threshold adjusts the spike response threshold with reference to potentially multiple time-dependent variables. When the model potential on the leaky integrator reaches the spike threshold, the model fires and a spike is generated. The model is not allowed to fire for 2ms after the last spike. This refractory period ensures to a certain extent that no singular bursting will occur.

Leaky Integrator:

$$\tau m \frac{dV}{dt} = -V + RI(t) \tag{1}$$

Based on the assumption of the non-resetting model potential, the threshold will increase when the model fires and decline afterward exponentially towards the resting value. Further explanation of this spiking behaviour is found in Section 3.

Spike Threshold:

$$\theta(t) = \sum_k H(t - t_k) + \omega \tag{2}$$

with

$$H(T) = \sum_{j=1}^{L} \alpha_j^{\frac{-t}{\tau_j}} \tag{3}$$

Generally, the MAT-Model is referred as MAT(L)-Model, where L describes different timescales. The implementation will be restricted to a MAT(2)-Model with 10ms and 200ms as timescales, also referred to as the MAT* Model. These are trial parameters that have proven to be performant for predictions (Kobayashi et al., 2009).

The following main functions are derived from the described MAT-Model, which need to be realized during the implementation phase:

1. Generate Input Currents

   Generation of artificial input currents, which are injected into the modelled neuron.

2. Get Model Potential

   Calculation of the model potential based on given input currents.

3. Get Spike Thresholds and Spike Threshold Variables.

   Determines the spike threshold and the spike threshold time dependent variables for modeling different types of neurons.

4. Predict Spikes

   Predicts spikes based on the injected input currents and the spike threshold.

5. Evaluate Model Prediction

   Evaluate the accuracy in the form of a coincidence factor of model-predicted spikes against ground truth neuron firing data.

With the implementation of the described main functions the realization of a fully functional MAT-Model is pursuant. The implemented model will be used for further evaluation.

2.2 Evaluating the Results

The second main objective of the project is the evaluation of the results, which are based on the implemented model according to the previous section. This evaluation is divided into the two following major tasks:

1. Evaluation of the Spike Prediction Performance

   An essential part of the evaluation is the analysis of the spike prediction performance of the implemented MAT-Model. This evaluation requires a set of reference data, such evaluation requires a set of ground-truth reference data. This reference data enables a comparison with the observed data from the implemented model, which allows a qualitative reflection of the implemented model regarding the spike prediction performance.

2. Evaluation of different Types of Neurons

   Given the statement that the original MAT-Model can represent the behaviour of different types of neurons, the implemented model is examined to determine if it can also represent these types of neurons. For the fulfillment of this requirement, evaluations of two neurons for each of the following types are targeted:

   - Regular-Spiking Neurons
   - Fast-Spiking Neurons

   These two evaluations are considered as sufficient to represent the efficiency of the implemented model with respect to the limited scope of this project.

The evaluation of the MAT model is completed by carrying out an evaluation of the general prediction functionality and considering multiple types of neurons. The results of this analysis allow a sufficient statement about the quality of the implemented model.

3 Phase Space Analysis

Before exploring the model's implementation, we first describe briefly the qualitative behavior of the system via phase space analysis. Although a phase space analysis cannot depict spike dynamics per se given the threshold-conditional spike generation, we can still analyze the dynamical behavior of the model via the adaptive threshold equations 2 and 3. Assuming the two timescale (L=2) MAT* model, the adaptive threshold $\theta(t) - w$ decomposes into variables ϕ_1 and ϕ_2 via equation 3.

$$\begin{aligned}\theta(t) - w &= \sum_k \; H(t - t_k) \\ \phi_1 &= \sum_k \; \alpha_1 \; exp(-t - t_k/\tau_1) \\ \phi_2 &= \sum_k \; \alpha_2 \; exp(-t - t_k/\tau_2)\end{aligned}$$

Where τ_1, τ_2, and k represent the first and second time constants and the kth spike time, respectively. Accordingly, we can derive the following differential equations for the two variables (cf. Yamauchi et al., 2011). By solving these equations, we can integrate with respect to t to obtain the function of model state (ϕ_1, ϕ_2), relative to the derived variables.

$$d\phi_1/dt = -\,\phi_1/\tau_1 \qquad\qquad d\phi_2/dt = -\,\phi_2/\tau_2$$

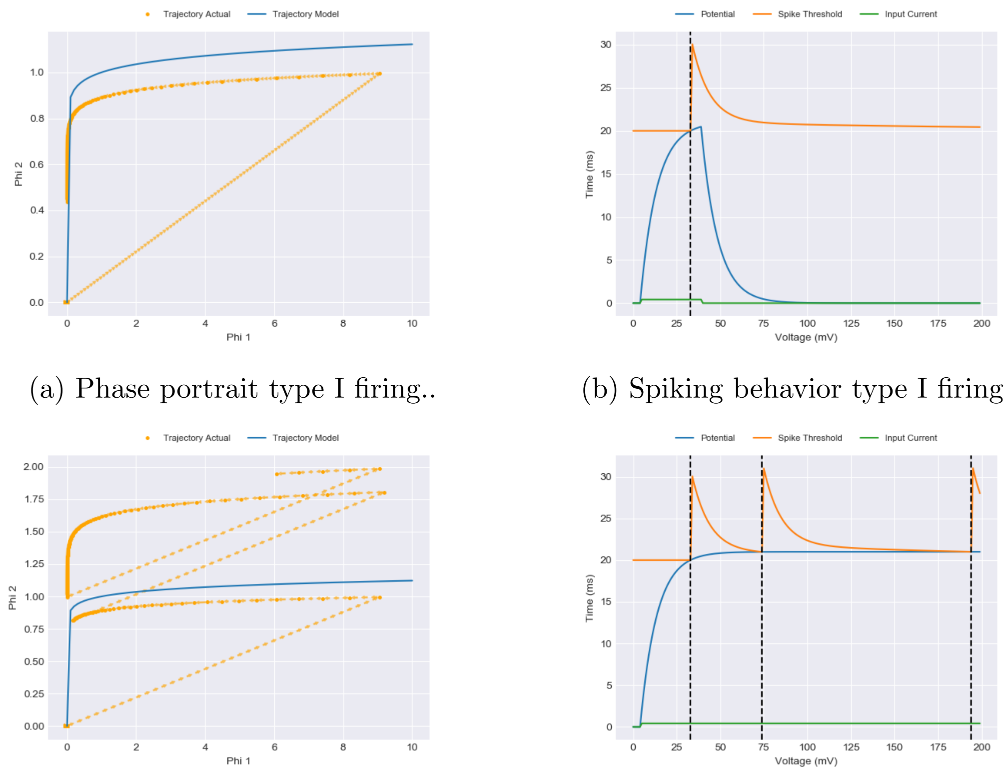
$$\frac{\tau_1 \cdot (d\phi_1/dt)}{\phi_1} = -\,1 \qquad\qquad -1 = \frac{\tau_2 \cdot (d\phi_2/dt)}{\phi_2}$$

$$\int \frac{\tau_1 \cdot (d\phi_1/dt)}{\phi_1} = \int -1 \qquad\qquad \int -1 = \int \frac{\tau_2 \cdot (d\phi_2/dt)}{\phi_2}$$

$$\tau_1 \ln \phi_1 \;=\; c \;=\; \tau_2 \ln \phi_2$$

$$\therefore\; \phi_1 \cdot \tfrac{\tau_1}{\tau_2} = \phi_2$$

Where c is an arbitrary integration constant.



(a) Phase portrait type I firing..

(b) Spiking behavior type I firing..

(c) Phase portrait bursting.

(d) Spiking behavior bursting..

Figure 1: Model dynamics for type I firing and bursting.

Assuming $\tau_1 = 10$ and $\tau_2 = 200$ per the default values chosen by (Kobayashi et al., 2009), by plotting the derived equation we obtain a phase portrait depicting model state trajectory as a function of the adaptive threshold variables. Figures 1a and 1b depict the phase portrait for type I firing, including the actual trajectory produced by our implementation and the corresponding spike behavior. There exists a demonstrable stable fixed point at the origin, $(\phi_1, \phi_2) = (0,0)$ to which the model state returns following a spike in the case that the model potential shown in equation (1) exceeds the adaptive threshold.

Likewise, bursting dynamics are depicted with compounding state shifts following a spike, shown in Figures 1c and 1d. Following a spike, model state jumps according to $(\phi_1, \phi_2) \rightarrow (\phi_1+\alpha_1, \phi_2+\alpha_2)$ before returning to the origin. In summary, the MAT* model's behavior can be depicted in phase space via decomposition of the adaptive threshold equations into a single exponential function with one stable fixed point, modeling both bursting and singular firing patterns with respect to the adaptive spiking condition.

4 Technical Solution and Implementation

To ensure a structured software implementation approach, the implementation phase of this project is divided into two sub-modules. The definition of separate modules allows a more detailed analysis of the implementation tasks and efficient target tracking. In this project the defined modules for the implementation of the MAT-Model are the following:

1. Experiment Input

   Data from a selected experiment is required for the model implementation and evaluation. The loading and processing of the experiment data is the focus of the first module.

2. Model Prediction

   The second module is related to the prediction of the spiking behaviour based on the given input currents from the experimental data.

These two modules, their functions and main variables are shown in the swim lane diagram in figure 2. The implementation of the defined modules is examined in more detail in the following sub sections.

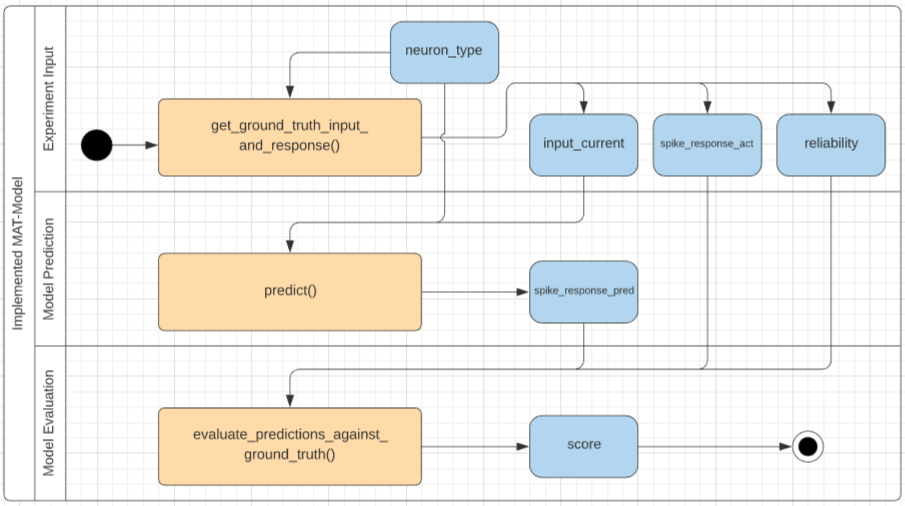


Figure 2: Swim lane diagram of implemented MAT-Model

4.1 Experiment Input

The required starting point for the implementation of the MAT-model is the input ground-truth neuron firing data from a selected experiment. The Quantitative Single-Neuron Modeling Competition 2009 (QSNMC) (Gerstner and Naud, 2009) is the selected experiment as baseline for the implementation and evaluation of the MAT-Model. The input currents and voltage responses for a regular spiking and a fast spiking neuron are used out of this experiment. To determine which neuron type should be used in the implemented MAT-Model, the string variable *neuron_type* must be defined manually.

Initially the input current and voltage responses for the defined neuron type are loaded into the model. Since the data set only provides a smaller set of voltage values than currents, it needs to be ensured that the same amount of data is considered for the further processing. In addition, the voltage response data includes multiple repetitions, where a repetition represents the output of the repeated experiment using the same input currents. Therefore, the input currents and voltage responses are handled in lists of the same length, in which the data are sorted to one another using a freely selected repetition. The new list for the voltage is assigned to the variable *spike_response_act* and describes the spike responses of the experimental data.

Before the list of input currents are assigned to the variable *input_current* the unit of the current must be converted, since the experimental data of the current are given in pA, whereas the model uses nA. It is necessary to mention that the given data provides a total of 600,000 input current values based on a experiment length of 60 seconds. This results in a time step of 0.1 ms per current injection. In addition, the variable *reliability* is calculated based on the voltage responses. The *reliability* describes the averaged coincidence factor that is gathered by comparing the spike trains of the given different repetitions and calculating their individual coincidence factors to each other.

The variable *input_current* serves as input for the module Model Prediction, whereas the variables *spike_response_act* and *reliability* are needed for the evaluation of the implemented model.

4.2 Model Prediction

To enable a comparison of the prediction with the ground truth data in the evaluation, the prediction functionality of the implemented MAT-Model will use the same set of input currents as the ground-truth data. In addition, the variable *neuron_type* is required within this module to determine the spike threshold time dependent vari-

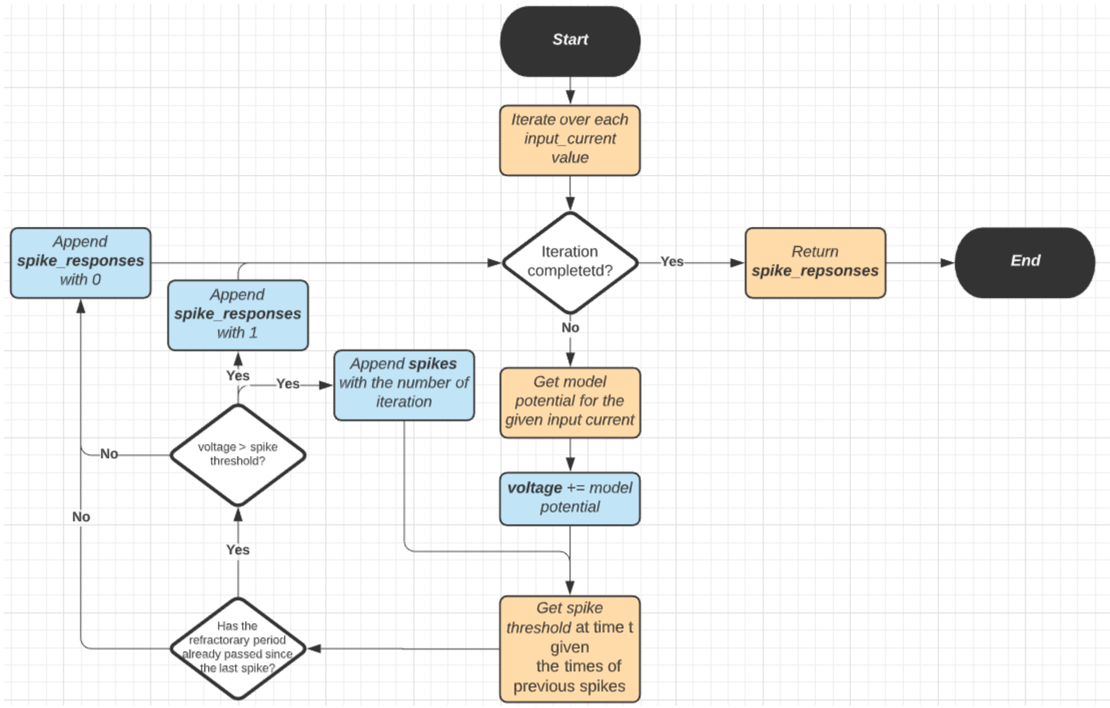


Figure 3: Flowchart of Model Prediction module

ables α_1, α_2 and ω. These variables are derived from (Kobayashi et al., 2009) based on their optimization on injected current data.

The main processes of the implemented Model Prediction module are shown in Figure 3. The model iterates over each value of the variable *input_current* and calculates the model potential for each value. These values are added up during the iteration, since the MAT-model does not reset the voltage value. For each time step t, the spiking threshold is calculated. The time step t corresponds to the moment of the respective input current injection and therefore allows allows representation as iteration across all input currents. The spike threshold is determined given all the times of previous spikes. As next step it will be checked if the defined refractory period is passed since the last spiking of the model, which is a given constraint from the MAT-Model and the related natural behaviour. With respect to the constraint of a 2ms refractory period, 20 iterations over the *input_current* need to pass by until the neuron is able to fire again. If the refractory period is not passed the *spike_response* list will be appended with a 0, indicating that the model hast not fired. If the refractory period following a spike is completed, the model checks if its potential exceeds the adaptive threshold. If the model potential is greater then the threshold the *spike_response* list will be appended with an 1, indicating that the neuron has fired. In addition, the list *spikes* will be appended with the current time step t, to enable the consideration of all previous spike times in the calculation of the spike threshold.

With the completion of the iteration over all input currents, the list *spike_response* represents the predicted spiking behaviour of the implemented Model Prediction module. This list will be used evaluation of the model.

5 Evaluation

We evaluate our implementation's performance with ground-truth neuron firing data from the QSNMC (2009). The competition offers four challenges for predicting spike times based on input current data and voltage responses measured for different neuron types. For our evaluations, we use only challenges A and B, which record the input currents and voltage responses for a regular-spiking and fast-spiking neuron, respectively.

5.1 Methodology

Spike time prediction is assessed via the coincidence factor Γ, as in (Kobayashi et al., 2009). The coincidence factor, summarized in equation (4), is also the performance evaluation metric used in the QSNMC.

$$\Gamma \;=\; \frac{N_{coinc} - \langle N_{coinc} \rangle}{N_{data} + N_{model}} \cdot \alpha \tag{4}$$

where N_{data} and N_{model} denote the number of spikes in the data and model spike trains, respectively. N_{coinc} refers to the number of coincident spikes between the model and data spike trains which co-occur within a time window Δ. In our evaluations, we use $\Delta = \pm 2ms$ and $\Delta = \pm 4ms$ as done by (Kobayashi et al., 2009), and (Gerstner and Naud, 2009), respectively. $\langle N_{coinc} \rangle$ denotes the number of expected coincidences generated by a homogeneous Poisson process with the same firing rate v as the model spike train, computed as $\langle N_{coinc} \rangle \;=\; 2v\Delta N_{data}$. The factor $\alpha = \frac{2}{1-2v\Delta}$ normalizes the coincidence factor to a maximum of 1. A homogeneous Poisson process with the same number of spikes as the ground-truth data will yield $\Gamma = 0$, which is the chance value. Consequently, a perfect score is $\Gamma = 1$. For further details please see (Jolivet et al., 2008).

When implementing the aforementioned evaluation metric we impose several conversions and simplifying assumptions given the specifications of the QSNMC (2009) data and our implementation. Firstly, the QSNMC (2009) input current data is recorded in pA at a time-step of $10t = 1ms$, where t denotes a single time-step. As our implementation assumes a $t = 1ms$ time-step, we adjust the original MAT model hyperparameters τ_1, τ_2, τ_m, and the refractory period to 100, 2000, 50, and 20 respectively, that they be adjusted to the modified ground-truth data time-step. These hyperparameter values remained constant across all evaluation trials documented below. Additionally, we convert the input current values to nA per the original MAT model assumptions.

Secondly, we impose a recovery period on the ground-truth data spike-counting which stipulates that the number of potential spikes be analogous to those predicted by the model. The QSNMC (2009) evaluation regulations state that a spike time is defined as "the time at which the voltage recording is crossing 0 mV". (Gerstner and Naud, 2009) As the voltage may remain above 0mv for several time-steps following a spike, we impose the additional constraint which defines a ground-truth spike as the time-steps for which the voltage is crossing 0mV within a refractory period of 2ms, as imposed on the model implementation itself.

Furthermore, when computing the model firing rate v for calculating the number of predicted coincident spikes of a homogeneous Poisson process, firing rate is computed dynamically as the number of spikes occurring within a 5ms window after the first predicted spike. This methodology for computing v avoids outlier predictions resulting from noise and lengthy periods of inactivity in the QSNMC (2009) data. The 5ms time window for calculating firing rate is selected as a standard value suggested by (Gerstner et al., 2014).

Finally, when evaluating on the ground-truth data, we remove the first 150,000 time-steps as these data do not contain any spikes and only increase computation time.

Model performance Γ on regular-spiking and fast-spiking neuron data from the QSNMC (2009) is summarized in the tables below. The tested configurations of parameter values were identified from the original MAT paper, cf (Kobayashi et al., 2009) p.2. For each set of hyperparameters we record the Γ value when the coincidence precision Δ is set to $\pm$ 2ms and $\pm$ 4ms, per the original MAT model evaluations, and the QSNMC scoring procedures, respectively.

5.2 Regular-Spiking Neuron

Model performance on the QSNMC data collected from a regular-spiking neuron using the different parameter configurations provided in (Kobayashi et al., 2009) can be found in the table below.

The table demonstrates that the model performs worst when using the suggested

Parameter Values	$\Delta = \pm 2ms$	$\Delta = \pm 4ms$
RS: $\alpha_1 = 37, \alpha_2 = 2, w = 19$	0.322	0.617
RS*: $\alpha_1 = 200, \alpha_2 = 3, w = 19$	**0.467**	**0.700**
IB: $\alpha_1 = 1.7, \alpha_2 = 2, w = 26$	0.459	0.555
FS: $\alpha_1 = 10, \alpha_2 = 0.002, w = 11$	0.207	0.263
CH: $\alpha_1 = -0.5, \alpha_2 = 0.4, w = 26$	0.353	0.399

Table 1: Model performance on regular-spiking neuron data

parameters of (Kobayashi et al., 2009), whereas the provided parameters for the intermittent-bursting neuron provide the best results. The parameters RS* were manually determined to provide optimal results.

5.3 Fast-Spiking Neuron

Model performance on the QSNMC data collected from a fast-spiking neuron using the different parameter configurations provided in (Kobayashi et al., 2009) can be found in the table below.

Parameter Values	$\Delta = \pm 2ms$	$\Delta = \pm 4ms$
RS: $\alpha_1 = 37, \alpha_2 = 2, w = 19$	0.573	0.883
IB: $\alpha_1 = 1.7, \alpha_2 = 2, w = 26$	0.411	0.548
FS: $\alpha_1 = 10, \alpha_2 = 0.002, w = 11$	**0.275**	**0.358**
CH: $\alpha_1 = -0.5, \alpha_2 = 0.4, w = 26$	0.411	0.487

Table 2: Model performance on fast-spiking neuron data

The model parameters for fast-spiking neurons did not yield the best predicting performance out of all tested parameters on the given data for fast-spiking neurons; the parameters for regular-spiking yielded significantly better results. Moreover, the model parameters for regular-spiking neurons predicted the behavior of the fast-spiking neuron data better ($\Gamma_{2ms} = 0.573$) than the behavior of the regular-spiking neuron data ($\Gamma_{2ms} = 0.322$).

5.4 Discussion

The Γ values for the different parameter configurations across the two neuron types suggest that the model is marginally effective at predicting spikes, but that the values determined by (Kobayashi et al., 2009) cannot be effectively generalized to the QSNMC (2009) data. In fact, for the regular-spiking QSNMC data, the suggested regular-spiking neuron parameters produce some of the worst performance.

As expected, the model generally performs better when $\Delta = 4$ms.

When comparing the implementation's performance to that of the original MAT model for a regular-spiking neuron, we can say the original MAT model vastly outperforms our implementation with a Γ value of 0.89 ± 0.21 with $\Delta = 2$ms. However, as this performance is achieved via tuning parameters to a different dataset, it is expected that our implementation will perform worse. This hypothesis is corroborated by the best performing set of parameter values for all Δ values on the regular-spiking data, RS*, a parameter set which was manually-determined via experimentation. It can be assumed that dynamic fitting of model parameters to the ground-truth data would yield improved Γ performance, but such procedures are outside the scope of this project.

Qualitative evaluation of the model's spike predictions further suggests that additional hyperparameter tuning will yield competitive performance. As seen in Figure 4, many ground-truth spikes (red) are correctly predicted (black) or very closely-predicted.

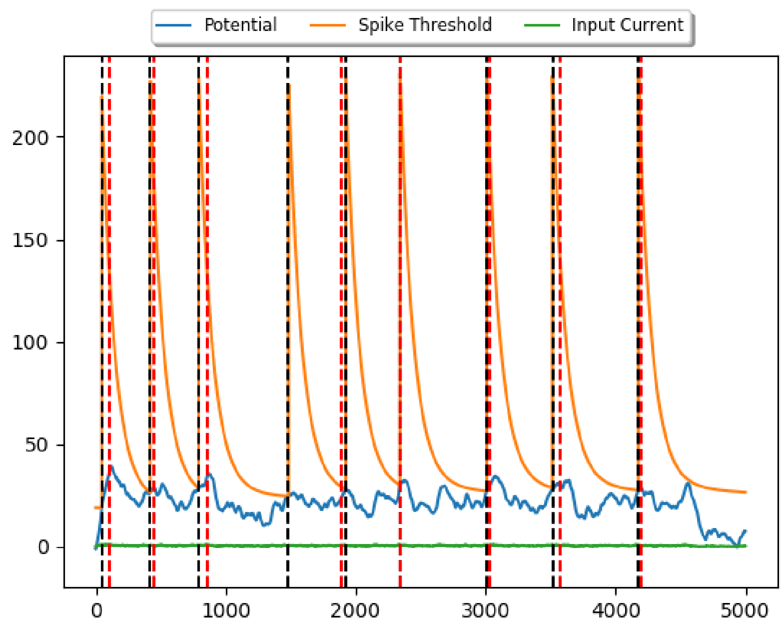


Figure 4: Model predictions (black) and ground-truth (red) spikes for regular-spiking neuron data. Slice: 25-30ms.

Similar to its performance on regular-spiking neuron data, when predicting on the fast-spiking data the model does not achieve competitive performance compared

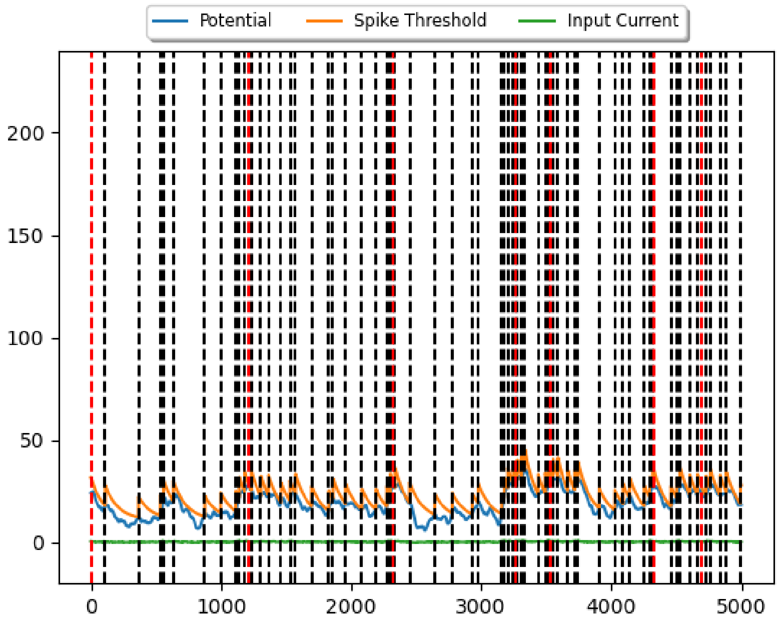


Figure 5: Model predictions (black) and ground-truth (red) spikes for fast-spiking neuron data. Slice: 60-65 ms.

to the implementation from the original paper. Moreover, the parameters for modelling fast-spiking neurons provided by (Kobayashi et al., 2009) yielded the worst results on our test data out of all the parameter configurations. The recorded fast-spiking neuron does not fire as "fast" as the modelled one (see Figure 5). This leads to the hypothesis that the neuron Kobayashi and colleagues used for parameter optimization exhibited firing behavior of higher frequency than the fast-spiking neuron from the QSNMC (2009) dataset.

It must be admitted that there is a possibility of our evaluation procedure being flawed, given that we cannot assess the accuracy of the coincidence factor as-implemented without access to the original data of (Kobayashi et al., 2009). For similar reasons, the evaluations are limited in the conclusions which may be drawn from them because we have access only to the publicly-available QSNMC (2009) challenge data and not other data sources which may be more relevant to the types of neurons tested in the original paper.

6 Conclusion

Before concluding we compare our main goals with achieved results to evaluate the success of the project. The aforementioned goals are summarized below.

1. Implementing the MAT-Model
2. Evaluation of the Results

On the implementation side, we have achieved all relevant requirements including the ability to automatically generate synthetic input currents, calculate model potential based off input currents and predict spikes based on injected input currents and implemented the adaptive spike threshold. We have also implemented the evaluation of predicted spikes against ground-truth data via the coincidence factor Γ.

Nevertheless, the model's evaluation is limited by the incompatibility between the acquired ground-truth data and optimized parameters provided by (Kobayashi et al., 2009). Consequently, evaluations indicate that the model's predictive performance is vastly inferior to that of the original paper.

The main goals assessed in the problem description of this work have been achieved to a satisfactory degree, although evaluation indicates that our model is only partially-effective in comparison to the reference implementation.

References

Jolivet, R., Kobayashi, R., Rauch, A., Naud, R., Shinomoto, S., & Gerstner, W. (2008). A benchmark test for a quantitative assessment of simple neuron models. *Journal of neuroscience methods*, *169*, 417–24. https://doi.org/10.1016/j.jneumeth.2007.11.006

Gerstner, W., & Naud, R. (2009). How good are neuron models? *Science*, *326*(5951), 379–380. https://doi.org/10.1126/science.1181936

Kobayashi, R., Tsubo, Y., & Shinomoto, S. (2009). Made-to-order spiking neuron model equipped with a multi-timescale adaptive threshold. https://www.frontiersin.org/articles/10.3389/neuro.10.009.2009/full

Yamauchi, S., Kim, H., & Shinomoto, S. (2011). Elemental spiking neuron model for reproducing diverse firing patterns and predicting precise firing times. *Frontiers in computational neuroscience*, *5*, 42. https://doi.org/10.3389/fncom.2011.00042

Gerstner, W., Kistler, W., Naud, R., & Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. https://doi.org/10.1017/CBO9781107447615