STN-GPe Network Simulation Models: Weight-Based and Conductance-Based Approaches

Overview

This document details two computational models developed to simulate the dynamics of the subthalamic nucleus (STN) and the external segment of the globus pallidus (GPe) within the basal ganglia. These models aim to replicate the neural oscillatory behaviors observed in normal and Parkinsonian states and to assess the impact of Deep Brain Stimulation (DBS) on these dynamics.

---

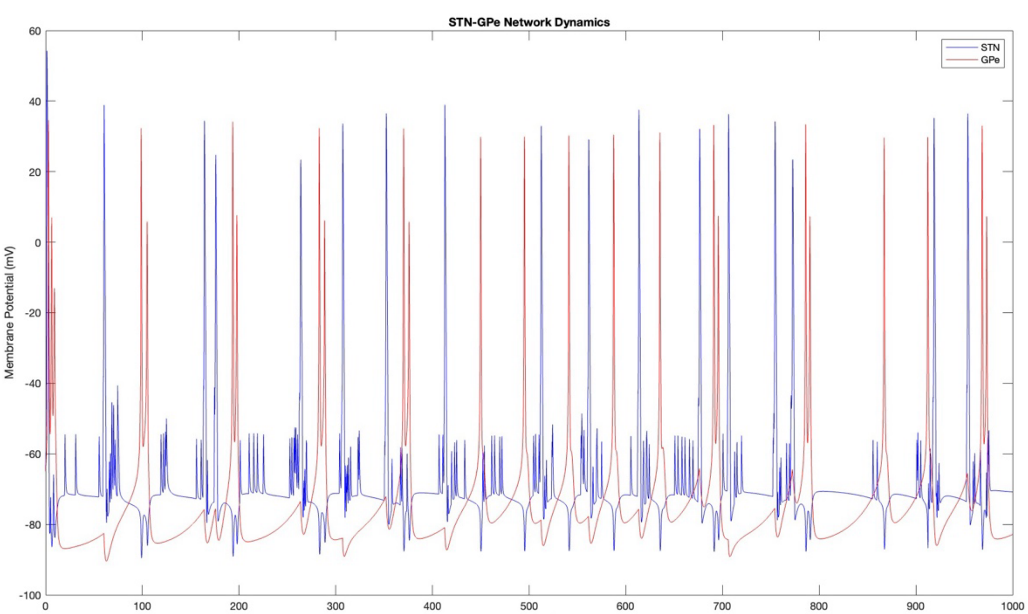
1. Closed Loop: Weight-Based Model

1.1 Description

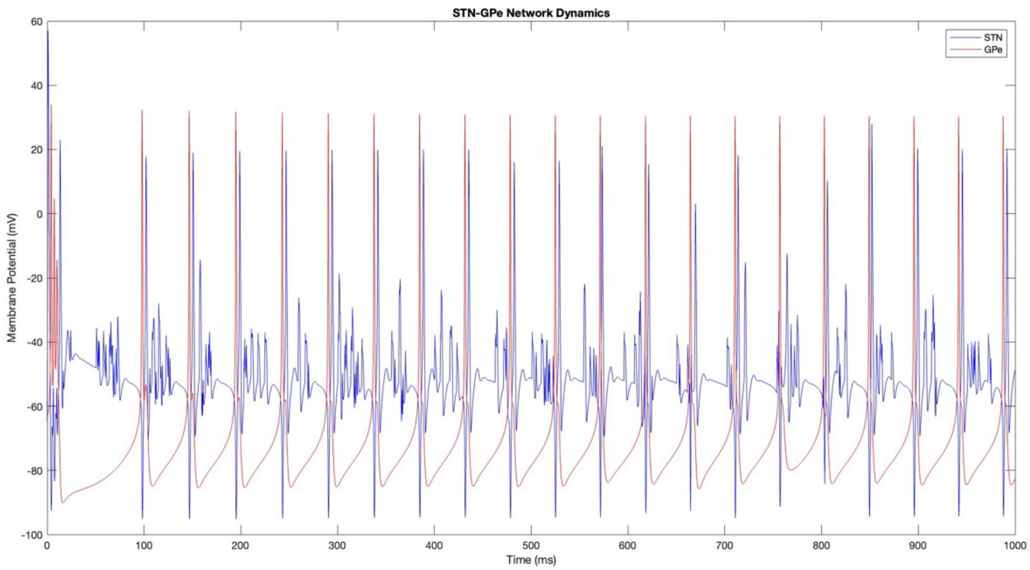
The weight-based model employs a simplified representation of synaptic interactions between **STN** and **GPe** neurons. Synaptic **weights** encapsulate the combined effects of **synaptic conductance, neurotransmitter release probabilities,** and **receptor dynamics.** This model is particularly useful for exploring the synchronization phenomena between **STN** and **GPe** neurons under varying **synaptic strengths**.

1.2 Main Program and Supporting Files

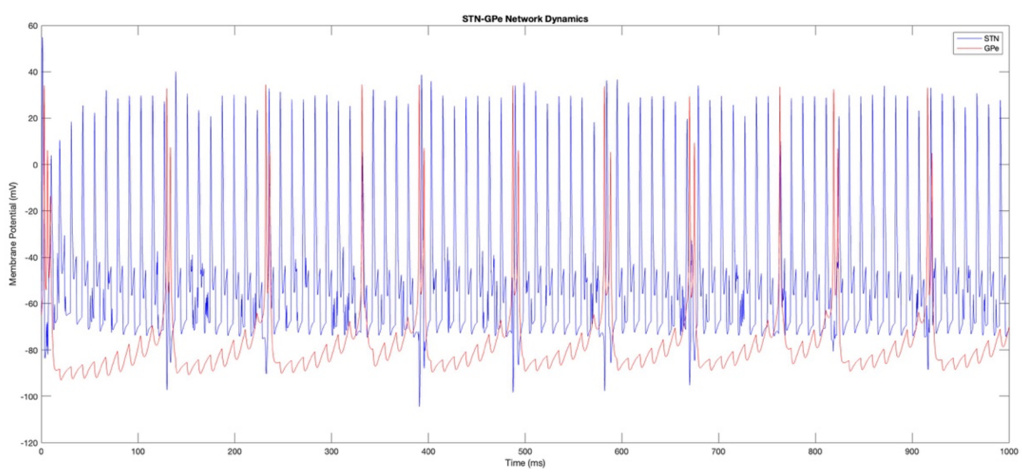
- **Main Program**: STN_GPe_WeightBased.m
- **Supporting Functions**:
  - generate_beta_poisson_bursts.m: Generates cortical beta-band input as Poisson-distributed bursts.
  - STN_Reduced_Step.m: Updates the state of the STN neuron using a reduced-order model.
  - GPe_Reduced_Step.m: Updates the state of the GPe neuron using a reduced-order model.
- **Results:**
  - Normal Weight-Based: Depicts STN and GPe activity under normal conditions.



- Parkinson Weight-Based without DBS: Shows the effect of DBS in Parkinsonian conditions.



- Parkinson Weight-Based with DBS: Illustrates Parkinsonian dynamics without DBS.



- Highlights code sections for parameter modifications

```
%% Vary the value from 0.4 to 1.5 to see in sync action
W_GPe2STN = 0.4; % Normal functioning
%W_GPe2STN = 1.5; % Parkinson Condition

for i = 1:nSteps-1
        % Compute synaptic currents
        I_syn_STN = g_GPe_STN * s_GPe_STN(i) * (V_STN(i) - E_GPe_STN);
        I_syn_GPe = g_STN_GPe * s_STN_GPe(i) * (V_GPe(i) - E_STN_GPe);
        I_dbs(i) = scale * V_GPe(i) * ((mod(floor(t(i)/(2*pw)), 2) == 0) - 0.5) * 2;

        % Total injected currents including synaptic inputs
        Iinj_STN(i) = I_syn_STN;
        Iinj_STN(i) = Ictx(i) - W_GPe2STN * V_GPe(i);
        %Iinj_STN(i) = Ictx(i) - W_GPe2STN * V_GPe(i) + I_dbs(i);  % Excitatory cortex + inhibitory GPe
        Iinj_GPe(i) = I_syn_GPe;
```

1.3 Simulation Procedure

1. **Initialization**: Set initial membrane potentials and synaptic weights.
2. **Input Generation**: Use generate_beta_poisson_bursts.m to simulate cortical input.
3. **Time Loop**: For each time step:
   - Compute synaptic currents based on current weights.
   - Update neuron states using STN_Reduced_Step.m and GPe_Reduced_Step.m.
   - Adjust synaptic weights if plasticity is modeled.

1.4 Modifying Conditions

- **Normal Condition**: Set synaptic weight from GPe to STN (W_GPe2STN) to a baseline value (e.g., 0.4).
- **Parkinsonian Condition**: Increase W_GPe2STN to simulate enhanced inhibitory feedback (e.g., 1.5).
- **DBS Implementation**: Introduce a feedback mechanism that records GPe membrane potential, inverts it, scales it, and injects it into the STN to disrupt pathological synchrony.

Refer to Weight_Based.png for specific code sections to modify these parameters.

---

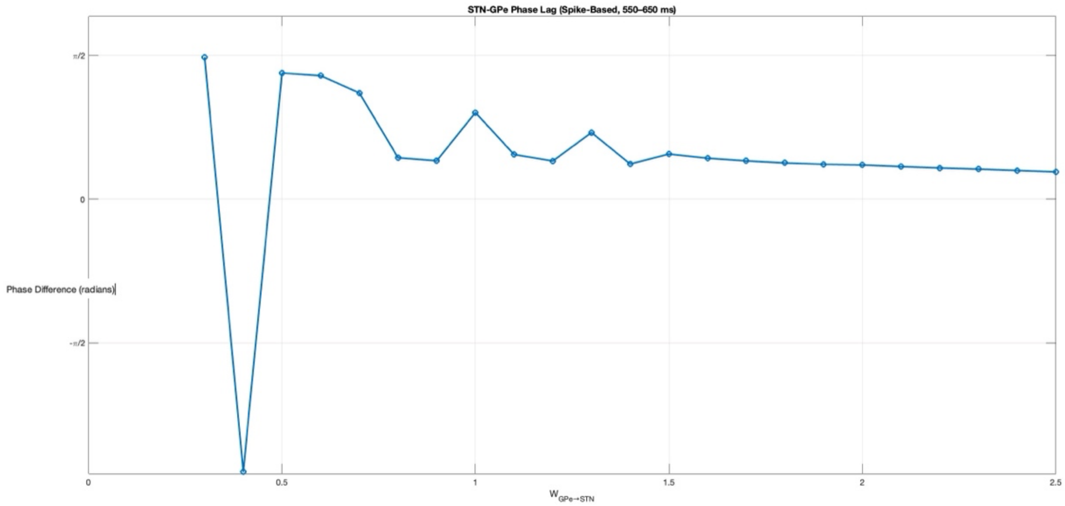
2. Weight Sweep Analysis

2.1 Description

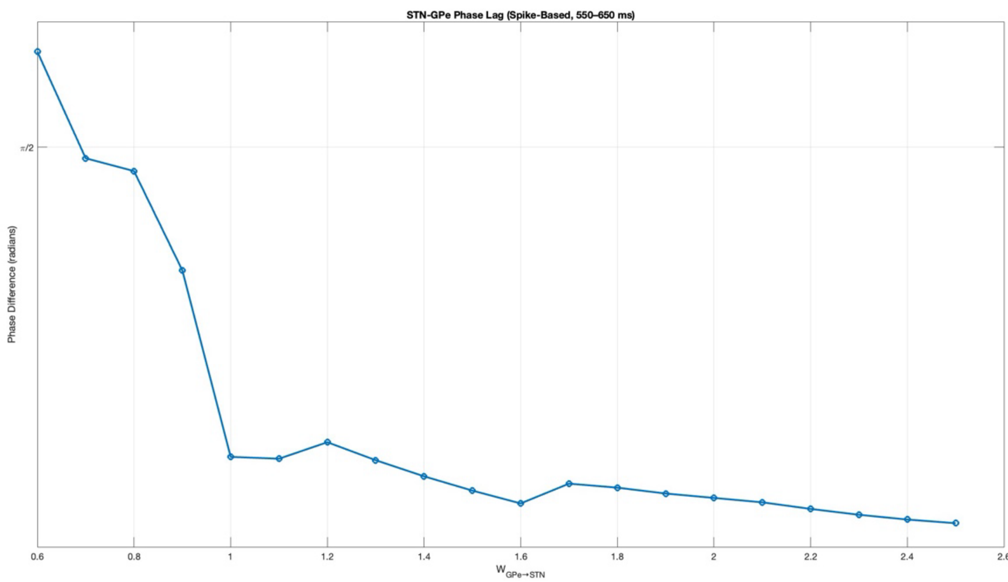
The weight sweep analysis systematically varies the synaptic weight from GPe to STN to study its effect on the phase relationship between STN and GPe neurons. This approach helps identify critical thresholds where synchronization emerges, mimicking Parkinsonian dynamics.

2.2 Main Program and Supporting Files

- **Main Program**: STN_GPe_WeightSweep.m
- **Supporting Functions**:
  - generate_beta_poisson_bursts.m: Generates cortical input.
  - phase_diff_STN_GPe_range.m: Calculates phase differences across weight values.
  - STN_Reduced_Step.m and GPe_Reduced_Step.m: Update neuron states.
- **Visualization Files**:
  - Phase_Weight_Sweep_1: shows phase differences with STN leading.



- Phase Weight Sweep 2: Shows phase differences with GPe leading.



2.3 Simulation Procedure

1. **Weight Range Definition**: Define a range of W_GPe2STN values (e.g., 0.4 to 2.5).
2. **Iterative Simulation**: For each weight value:
   - Run the simulation as per the weight-based model.
   - Compute phase differences using phase_diff_STN_GPe_range.m.

---

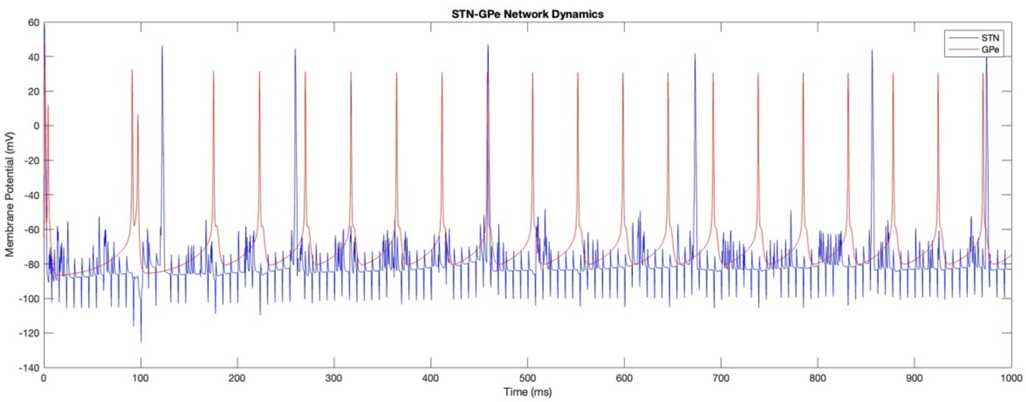
3. Open Loop: Conductance-Based Model

3.1 Description

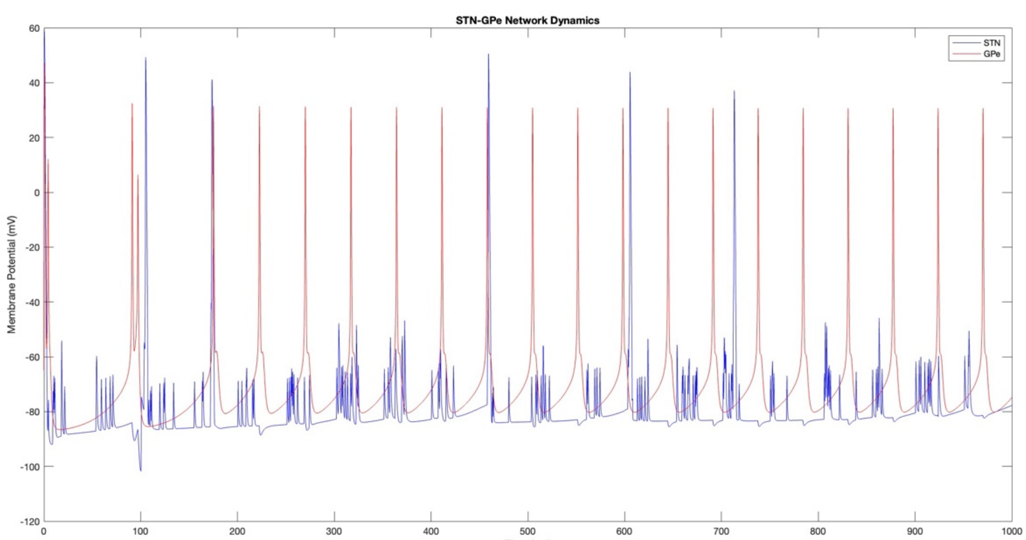
The conductance-based model provides a biophysically detailed representation of STN and GPe neurons, incorporating ion channel dynamics and conductance-based synaptic interactions. This model is suited for studying the effects of high-frequency DBS on pathological synchrony.

3.2 Main Program and Supporting Files

- **Main Program**: STN_GPe_Conductance.m
- **Supporting Functions**:
  - generate_beta_poisson_bursts.m: Generates cortical input.
  - generate_biphasic_dbs.m: Creates biphasic DBS waveforms.
  - STN_Reduced_Step.m and GPe_Reduced_Step.m: Update neuron states.
- **Visualization Files**:
  - Parkinson with DBS: Depicts the effect of DBS in Parkinsonian conditions.



  - Parkinson without DBS: Shows Parkinsonian dynamics without DBS.



- Highlights code sections for parameter modifications.

```
%% Vary the conductance to achieve sync action
%g_GPe_STN = 0.4;
g_GPe_STN = 1.2;

for i = 2:nSteps-1
    % Compute synaptic currents


    % Total injected currents including synaptic inputs and DBS
    I_syn_GPe_to_STN = g_GPe_STN * s_GPe_STN(i) * (V_STN(i) - E_GPe_STN);  % Conductance-based

     Iinj_STN(i) = Ictx(i) + I_syn_GPe_to_STN;
    %Iinj_STN(i) = Ictx(i) + I_syn_GPe_to_STN + I_dbs(i);  %% inject current here
```

3.3 Simulation Procedure

1. **Initialization**: Set initial membrane potentials, conductances, and synaptic variables.
2. **Input Generation**:
   - Use generate_beta_poisson_bursts.m for cortical input.
   - Use generate_biphasic_dbs.m to create DBS waveforms.
3. **Time Loop**: For each time step:
   - Compute synaptic currents based on conductance values.
   - Update neuron states using STN_Reduced_Step.m and GPe_Reduced_Step.m.
   - Apply DBS currents to the STN neuron.

3.4 Modifying Conditions

- **Normal Condition**: Set GPe to STN conductance (g_GPe_STN) to a baseline value (e.g., 0.4 mS/cm²).
- **Parkinsonian Condition**: Increase g_GPe_STN to simulate enhanced inhibitory feedback (e.g., 1.2 mS/cm²).
- **DBS Implementation**: Activate the DBS current by setting appropriate parameters in generate_biphasic_dbs.m.

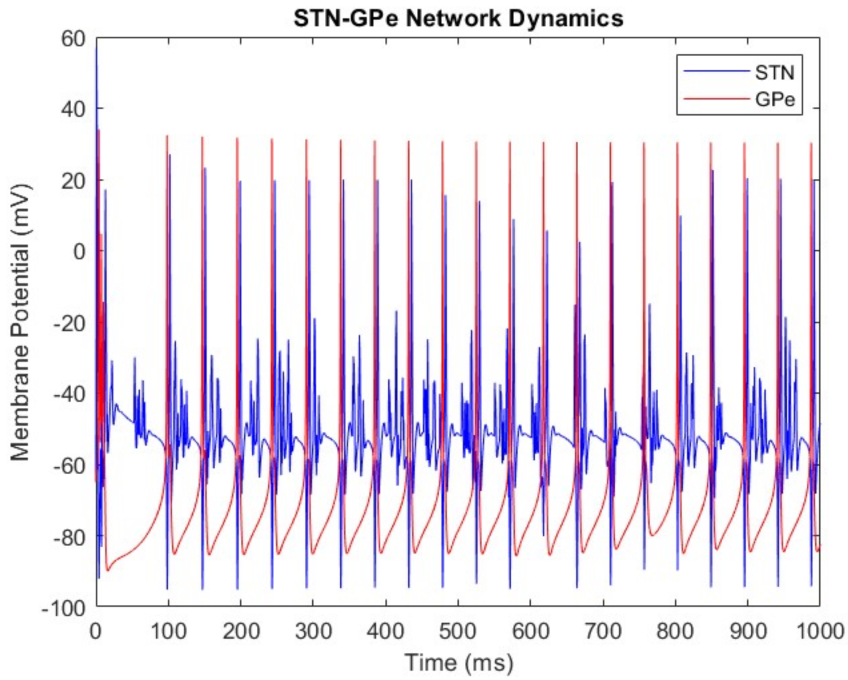
4. Heterogenous phase shifted Cortical Stimulation

This stimulation paradigm delivers charge-balanced, Poisson-distributed biphasic pulses to both STN and GPe with a deliberate 25 ms phase offset. At 20 Hz beta frequency (50 ms cycle), this corresponds to a 90° phase shift. STN receives input aligned with the cortical beta phase, while GPe receives the same input delayed. This temporal mismatch disrupts synchronous excitation-inhibition patterns that support beta-band resonance, effectively promoting network desynchronization.

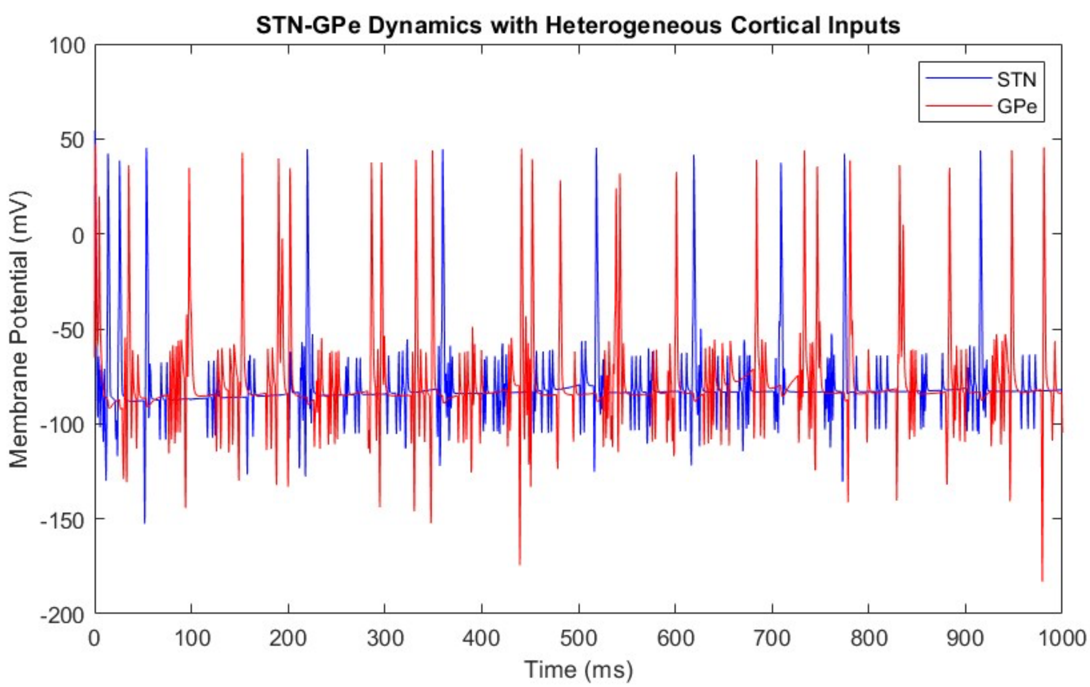
4.2 Main Program and Supporting Files

- **Main Program**: STN_GPe_heterogeneous_inputs.m
- **Supporting Functions**:
  - STN_Reduced_Step.m and GPe_Reduced_Step.m: Update neuron states.

  - Parkinson without DBS: Depicts the effect of DBS in Parkinsonian conditions.



- Parkinson with DBS: Depicts the effect of DBS in Parkinsonian conditions.



- ## 4.3 Simulation Procedure
- **Time Loop**: For each time step:
  - Compute synaptic currents based on conductance values.
  - Update neuron states using STN_Reduced_Step.m and GPe_Reduced_Step.m.
  - Apply DBS currents to the STN neuron.

4.4 Modifying Conditions

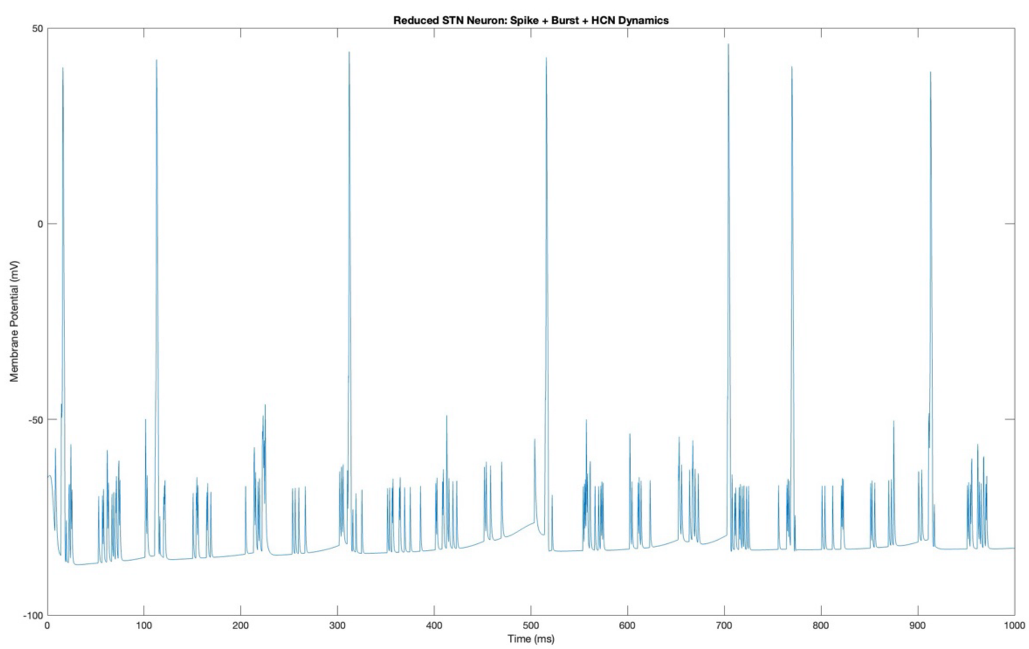
- **DBS Implementation**: Activate the DBS current by setting appropriate value of jitter (range: 0.5 – 1.2, optimal = 0.8)

5. Individual Neuron Simulations

5.1 STN Neuron-Only Simulation

This section focuses on simulating the intrinsic membrane dynamics of a single Subthalamic Nucleus (STN) neuron. The model uses reduced Hodgkin-Huxley-type equations to simulate ionic conductances and calcium concentration dynamics. The simulation applies an external beta-modulated Poisson input (generate_beta_poisson_bursts.m) to the STN neuron without any synaptic interactions with other neurons. This allows users to analyze the autonomous firing characteristics and response to input noise in both normal and parkinsonian regimes by adjusting ion conductances or current amplitudes.

- **Main Program**: simulate_STN_HH.m
- **Supporting Functions**:
  - generate_beta_poisson_bursts.m: Generates cortical input.

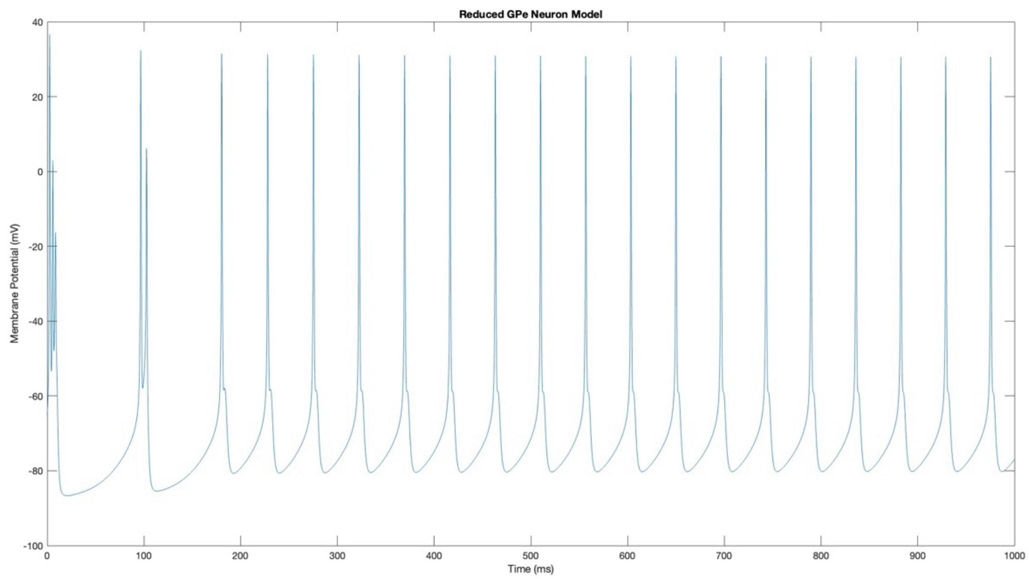


5.2 GPe Neuron-Only Simulation

This module isolates a single Globus Pallidus externus (GPe) neuron using its own set of biophysically realistic ion channels and gating variables. Like the STN simulation, it also excludes all synaptic inputs except **optional** noisy cortical-like input. This enables exploration of the GPe's intrinsic rhythmicity and response behavior. One can manipulate conductance values or inject different current profiles to study effects under baseline or pathological dopamine-depleted conditions.

- **Main Program**: simulate_GPe_HH.m

- **Supporting Functions**:
  - generate_beta_poisson_bursts.m: Generates cortical input.



4. Summary of Approaches

Refer to Conductance_Based.png for specific code sections to modify these parameters.

Model Type	Synaptic Representation	DBS Implementation	Modifiable Parameters	Use Cases
Weight-Based	Scalar synaptic weights	Closed-loop feedback	W_GPe2STN	Exploring sync thresholds
Conductance-Based	Biophysical conductances	Open-loop high-frequency	g_GPe_STN, DBS parameters	DBS effects on pathological dynamics
Heterogenous phase shifted Cortical Stimulation	Biophysical conductances	Open-loop high-frequency	jitter	Desync based on Poisson distributed noise

5. Interpretation of Results

- **Normal Condition**: STN and GPe neurons exhibit asynchronous firing patterns.
- **Parkinson Condition**: Enhanced inhibitory feedback leads to synchronized oscillatory activity.
- **DBS Application**: Disrupts pathological synchrony, restoring asynchronous firing patterns.