Contents

```
function dydt = ode_function(t, y)
```

```
    % Extract the variables from the y array
    V = y(1);
    m = y(2);
    h = y(3);
    m_NaP = y(4);
    h_NaP = y(5);
    n = y(6);
    m_T = y(7);
    h_T = y(8);
    m_P = y(9);
    m_N = y(10);
    h_N = y(11);
    z_SK = y(12);
    m_A = y(13);
    h_A = y(14);
    m_H = y(15);
    Ca_i = y(16);

   % Hypoglossal Motoneuron - Parameter Values
    g_max_Na = 0.7; % in micro siemens
    g_max_NaP = 0.05; % in micro siemens
    g_max_K = 1.3; % in micro siemens
    g_max_leak = 0.005; % in micro siemens
    g_max_T = 0.1; % in micro siemens
    g_max_P = 0.05; % in micro siemens
    g_max_N = 0.05; % in micro siemens
    g_max_SK = 0.3; % in micro siemens
    g_max_A = 1; % in micro siemens
    g_max_H = 0.005; % in micro siemens
    E_Na = 60; % Nernst potential for Na (60 mV)
    E_K = -80; % Nernst potential for K (-80 mV)
    E_leak = -50; % Nernst potential for Leak? (-50 mV)
    E_Ca = 40; % Nernst potential for Ca (40 mV)
    E_H = -38.8; % Nerst potential for Cations (-38.8 mV)
    K_1 = -0.0005; % I don't know what this is but its units are M/nC they may need to be adjusted to reflect SI units
    K_2 = 0.04; % ms^-1
    C_m = 0.04; % Membrane capacitance (0.040 nF)
```

```
Not enough input arguments.

Error in ode_function (line 3)
    V = y(1);
```

Sodium Current: I_Na

```
    m_inf = 1 / (1 + exp(-(V + 36) / (8.5)));
    tau_m = 0.1;
    h_inf = 1 / (1 + exp((V + 44.1) / 7));
    tau_h = (3.5 / ((exp((V + 35) / 4)) + (exp(-(V + 35) / 25)))) + 1;
```

Persistent Sodium Current: I_NaP

```
m_NaP_inf = 1 / (1 + exp(-(V + 47.1) / 4.1));
tau_m_NaP = 0.1;
h_NaP_inf = 1 / (1 + exp((V + 65) / 5));
tau_h_NaP = 150;
```

Delayed-Rectifier Current: I_K

```
n_inf = 1 / (1 + exp(-(V + 30) / 25));
tau_n = (2.5 / ((exp(V + 30) / 40) + (exp(-(V + 30) / 50)))) + 0.01;
```

LVA Calcium Current: I_T

```
m_T_inf = 1 / (1 + exp(-(V + 38) / 5));
tau_m_T = (5 / ((exp((V + 28) / 25)) + (exp(-(V + 28) / 70)))) + 2;
h_T_inf = 1 / (1 + exp((V + 70.1) / 7));
tau_h_T = (20 / ((exp((V + 70) / 65)) + (exp(-(V + 70) / 65)))) + 1;
```

HVA Calcium Current: I_N

```
m_N_inf = 1 / (1 + exp(-(V + 30) / 6));
tau_m_N = 5;
h_N_inf = 1 / (1 + exp((V + 70) / 3));
tau_h_N = 25;
```

HVA Calcium Current: I_P

```
m_P_inf = 1 / (1 + exp(-(V + 17) / 3));
tau_m_P = 10;
```

Calcium-Dependent Potassium Current: I_SK

```
z_SK_inf = 1 / (1 + (0.003 / Ca_i)^2);
tau_z_SK = 1;
```

Fast-Transient Potassium Current: I_A

```
m_A_inf = 1 / (1 + exp(-(V + 27) / 16));
tau_m_A = (1 / ((exp((V + 40) / 5)) + (exp(-(V + 74) / 7.5)))) + 0.37;
h_A_inf = 1 / (1 + exp((V + 80) / 11));
tau_h_A = 20;
```

Hyperpolarization-Activated Current: I_H

```
m_H_inf = 1 / (1 + (exp((V + 79.8) / 5.3)));
tau_m_H = (475 / ((exp((V + 70) / 11)) + (exp(-(V + 70) / 11)))) + 50;

% Calculate the currents based on the variables
I_Na = (g_max_Na) * (m^3) * (h) * (V - E_Na);
I_NaP = (g_max_NaP) * (m_NaP) * (h_NaP) * (V - E_Na);
I_K = (g_max_K) * (n^4) * (V - E_K);
I_leak = g_max_leak * (V - E_leak);
I_T = (g_max_T) * (m_T) * (h_T) * (V - E_Ca);
I_N = (g_max_N) * (m_N) * (h_N) * (V - E_Ca);
I_P = (g_max_P) * (m_P) * (V - E_Ca);
I_SK = 0 * (g_max_SK) * ((z_SK)^2) * (V - E_K);
I_A = g_max_A * m_A * h_A * (V - E_K);
I_H = g_max_H * m_H * (V - E_H);

%Applied Current
Iapp = 100;

% Compute the derivatives
dVdt = (1 / C_m) * (Iapp - (I_Na + I_NaP + I_K + I_leak + I_T + I_N + I_P + I_SK + I_A + I_H)); % equation for dV/dt
```

```
    dmdt = (m_inf - m) / tau_m; % equation for dm/dt
    dhdt = (h_inf - h) / tau_h; % equation for dh/dt
    dmdt_NaP = (m_NaP_inf - m_NaP) / tau_m_NaP; % equation for dm_NaP/dt
    dhdt_NaP = (h_NaP_inf - h_NaP) / tau_h_NaP; % equation for dh_NaP/dt
    dndt = (n_inf - n) / tau_n; % equation for dn/dt
    dmdt_T = (m_T_inf - m_T) / tau_m_T; % equation for dm_T/dt
    dhdt_T = (h_T_inf - h_T) / tau_h_T; % equation for dh_T/dt
    dmdt_P = (m_P_inf - m_P) / tau_m_P; % equation for dm_P/dt
    dmdt_N = (m_N_inf - m_N) / tau_m_N; % equation for dm_N/dt
    dhdt_N = (h_N_inf - h_N) / tau_h_N; % equation for dh_N/dt
    dzdt_SK = (z_SK_inf - z_SK) / tau_z_SK; % equation for dz_SK/dt
    dmdt_A = (m_A_inf - m_A) / tau_m_A; % equation for dm_A/dt
    dhdt_A = (h_A_inf - h_A) / tau_h_A; % equation for dh_A/dt
    dmdt_H = (m_H_inf - m_H) / tau_m_H; % equation for dm_H/dt
    dCa_idt = (K_1 * (I_T + I_N + I_P)) - (K_2 * Ca_i); % equation for dCa_i/dt

    % Pack the derivatives into a column vector
    dydt = [dVdt; dmdt; dhdt; dmdt_NaP; dhdt_NaP; dndt; dmdt_T; dhdt_T; dmdt_P; dmdt_N; dhdt_N; dzdt_SK; dmdt_A; dhdt_A; dmdt_H; dCa_idt];
```

```
end
```

Published with MATLAB® R2022b