TNM Exercise 3: DCM for EEG

Saurabh Vaishampayan (svaishampaya@student.ethz.ch)
Linus Rüttimann (rlinus@student.ethz.ch)
Elia Torre (etorre@student.ethz.ch)
Timothy Kurer (tkurer@student.ethz.ch)

Group **H**
18.04.2023

1 Code organisation

In the submission folder you will find the matlab script for ex 3.3.

3.1 Dynamic Equations

$$v(t) = \int_{-\infty}^{t} h(t-\tau)\sigma(\tau)d\tau = \int_{-\infty}^{t} HK(t-\tau)e^{-K(t-\tau)}\sigma(\tau)d\tau = HKe^{-Kt}\int_{-\infty}^{t}(t-\tau)e^{Kt}\sigma(\tau)d\tau =$$

$$= HKe^{-Kt}[t\int_{-\infty}^{t} e^{Kt}\sigma(\tau)d\tau - \int_{-\infty}^{t} \tau e^{Kt}\sigma(\tau)d\tau]$$

$$\dot{v}(t) = HK(-K)e^{-Kt}[t\int_{-\infty}^{t} e^{Kt}\sigma(\tau)d\tau - \int_{-\infty}^{t} \tau e^{Kt}\sigma(\tau)d\tau] + HKe^{-Kt}[\int_{-\infty}^{t} e^{Kt}\sigma(\tau)d\tau + te^{Kt}\sigma(\tau) - te^{Kt}\sigma(\tau)] =$$

$$= -Kv(t) + HKe^{-Kt}\int_{-\infty}^{t} e^{Kt}\sigma(\tau)d\tau$$

$$\ddot{v}(t) = -K\dot{v}(t) - HK^2e^{-Kt}\int_{-\infty}^{t} e^{Kt}\sigma(\tau)d\tau + HKe^{-Kt}e^{Kt}\sigma(\tau)$$

Therefore:

$$\ddot{v}(t) = -K\dot{v}(t) - K\dot{v}(t) - K^2v(t) + HK\sigma(\tau)$$

Which implies:

$$\ddot{v}(t) = HK\sigma(\tau) - 2K\dot{v}(t) - K^2v(t)$$

3.2 Coupled harmonic oscillator

3.2

a) $\ddot{x} = -f\dot{x} - K^2 x + u(t)$

Let $y = \dot{x}$

$\therefore \dot{y} = -fy - K^2 x + u(t).$

$\dot{x} = y.$

This can be rewritten in vector form by assuming $\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} y \\ -fy - K^2 x + u(t) \end{bmatrix}$$

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ -K^2 & -f \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ u(t) \end{bmatrix}$$

$\dot{\vec{x}} = A\vec{x} + \vec{u}(t)$ where $A = \begin{bmatrix} 0 & 1 \\ -K^2 & -f \end{bmatrix}$, $\vec{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$

and $\vec{u}(t) = \begin{bmatrix} 0 \\ u(t) \end{bmatrix}$

b) $u(t) = az(t)$

$$\ddot{z} = -f_z\dot{z} - K_z^2 z + u_z(t)$$

We are given that $z(t)$ is driven by $u_z(t)$

Following the same method as 3.2a), we can write linear 1^{st} order equation for $z(t)$ as

$$\frac{d}{dt}\begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_z^2 & -f_z \end{bmatrix}\begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ u_z(t) \end{bmatrix} \rightarrow b1$$

Also since $u(t) = az(t)$ we can write equations for $x(t)$ as :-

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K^2 & -f \end{bmatrix}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ az(t) \end{bmatrix} \rightarrow b2.$$

Combining b1 and b2 we get:-

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \\ z(t) \\ \dot{z}(t) \end{bmatrix}}_{\dot{\vec{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -K^2 & -f & a & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_z^2 & -f_z \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} x(t) \\ \dot{x}(t) \\ z(t) \\ \dot{z}(t) \end{bmatrix}}_{\vec{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ u_z(t) \end{bmatrix}}_{\vec{u}}$$

From the doubt asked on Moodle we ~~wer~~ assumed that no further specification for $u_z(t)$ is needed.

Hence $A_{41} = 0$. If $u_z(t)$ were coupled to $x(t)$ by some $u_z(t) = \cancel{bx}\ bx(t)$ then $A_{41} = b$ and $\vec{u}_{41} = 0$.

c) First we linearize $s(\vartheta) = \dfrac{1}{1+e^{-\tau\vartheta}} - \dfrac{1}{2}$ around $\vartheta = 0$

$$s(0) = \frac{1}{1+1} - \frac{1}{2} = 0,$$

$$s'(0) = \left[- \frac{(-\tau e^{-\tau\vartheta})}{(1+e^{-\tau\vartheta})^2} \right]_{\vartheta=0} = \frac{\tau}{4}.$$

$$\therefore\ s(\vartheta) \approx s(0) + \vartheta s'(0)$$

$$s(\vartheta) \approx \frac{\tau}{4}\vartheta(t).$$

We are given that in (3), $\sigma(t) = a s(\vartheta_z(t)) + u(t)$

$$\therefore\ \sigma(t) = a\frac{\tau}{4}\vartheta_z(t) + u(t)$$

$$\therefore\ \ddot{\vartheta}(t) = HK\left[a\frac{\tau}{4}\vartheta_z(t) + u(t)\right] - 2K\dot{\vartheta}(t) - K^2\vartheta(t).$$

Furthermore we are told that $\vartheta_z(t)$ also satisfies (3)

We assume $h_z(t) = H_z K_z t e^{-K_z t}$ (for generality).

$$\vartheta_z(t) = H_z K_z \sigma_z(t) - 2K_z\dot{\vartheta}_z(t) - K_z^2\vartheta_z(t)$$

Following ERP (Slide 57) we assume $\sigma_z(t) = a_z s_z(\vartheta(t))$ with $s_z(\vartheta) = \cancel{3}\dfrac{1}{1+e^{-\tau_z\vartheta}} - \dfrac{1}{2}$

$$\therefore\ \cancel{\vartheta_z(t) = H_z K_z} \{$$

$$\vartheta_z(t) = H_z K_z\left[a_z\frac{\tau_z}{4}\vartheta(t)\right] - 2K_z\dot{\vartheta}_z(t) - K_z^2\vartheta_z(t).$$

Similar to 3.2b we can write equations for $v(t)$ and $v_z(t)$ in matrix form as

$$\frac{d}{dt}\begin{bmatrix} v(t) \\ \dot{v}(t) \\ v_z(t) \\ \dot{v}_z(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K^2 & -2K & \frac{aHKr}{4} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{a_z H_z K_z r_z}{4} & 0 & -K_z^2 & -2K_z \end{bmatrix}\begin{bmatrix} v(t) \\ \dot{v}(t) \\ v_z(t) \\ \dot{v}_z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ HKu(t) \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} v(t) \\ \dot{v}(t) \\ v_z(t) \\ \dot{v}_z(t) \end{bmatrix}; \quad \vec{u} = \begin{bmatrix} 0 \\ HKu(t) \\ 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K^2 & -2K & \frac{aHKr}{4} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{a_z H_z K_z r_z}{4} & 0 & -K_z^2 & -2K_z \end{bmatrix}$$

$$\dot{\vec{x}} = A\vec{x} + \vec{u}$$

Analogy between neural state eqn for DCM and harmonic oscillator

For coupled HO we had

$$\ddot{z} = -f_z \dot{z} - k_z^2 z + u_z(t)$$

$$\ddot{x} = -f\dot{x} - k^2 x + az(t)$$

As mentioned in 3.2b, one can consider $u_z(t) = bx(t)$ (both forward & backward coupling).

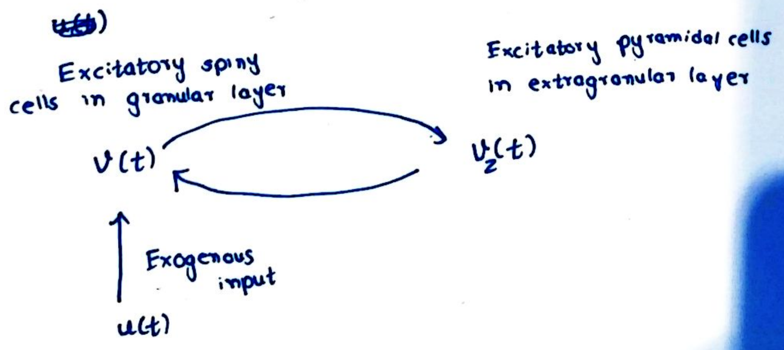
~~d) Connectivity diagram:-~~

In that case if one assumes that $x(t)$ corresponds with $v_z(t)$ and $z(t)$ corresponds with $v(t)$, then following correspondences hold

$$f \rightarrow 2K$$
$$K \rightarrow K$$
$$f_z \rightarrow 2K_z$$
$$K_z \rightarrow K_z$$
$$a \rightarrow \frac{a_z H_z K_z r_z}{4}$$
$$b \rightarrow \frac{aHKr}{4}$$

and $u_z(t) \rightarrow HKu(t)$.

d) Connectivity diagram:



3.3 Inference on network structure / parameter estimation

a

The following MATLAB code integrates the equation (11) over the interval $0 \leq t \leq 0.2s$:

```
kappa1 = 80;
kappa2 = 50;
f1 = 50;
f2 = 50;
a_f = 3000;
a_b = 1000;
iParam = 1;
mu = 0.05;
sigma = 0.01;

A = [0 1 0 0; -kappa1^2 -f1 a_f 0; 0 0 0 1; a_b 0 -kappa2^2 -f2];
C = [0; 0; 0; iParam];
dt = 0.001;
t = 0:dt:0.2;
u = normpdf(t,mu,sigma);
x = zeros(4,length(t));

for i=2:length(t)
    x(:,i) = x(:,i-1) + dt*(A*x(:,i-1)+C*u(i-1));
end

load('C:\Users\linus\Repos\translationalNeuromodeling\Exercise3\
    tn2023_ex3.mat');
plot(t,x);
legend('x1', 'x2', 'x3', 'x4');
xlabel('time [s]');
print(fullfile('C:\Users\linus\Repos\translationalNeuromodeling\
    Exercise3\Ex3_3a'),'-depsc');

%check if x and x_condition are equal:
norm(reshape(x-x_condition_1,1,[]))<eps % --> true
```

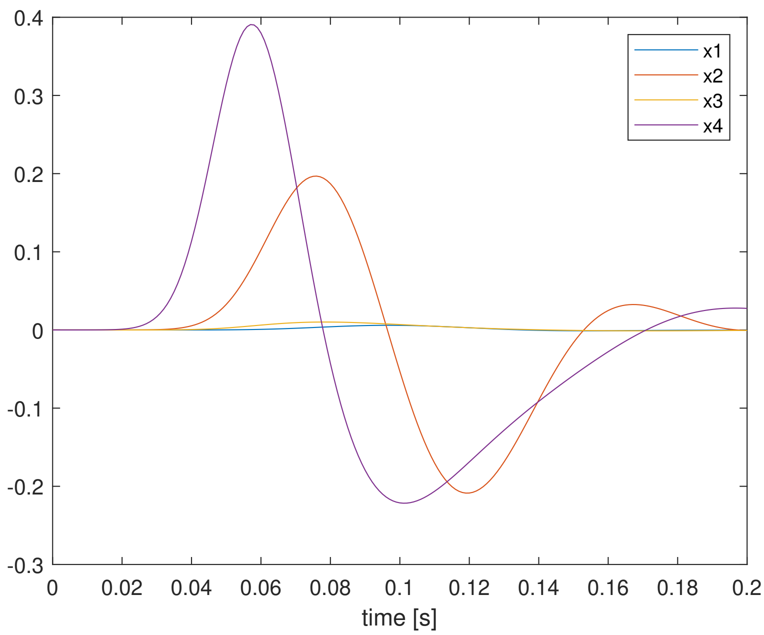


Figure 1: state time trajectory $x(t)$

b

The following MATLAB code performs a grid search for the four parameters:

```
kappa1_grid = 0:0.1:200;
kappa2_grid = 0:0.1:200;
a_f_grid = -1000:1:5000;
a_b_grid = -1000:1:5000;


varParams = ["kappa1", "kappa2", "a_f", "a_b"];

for iParam=1:4 % loop throup parameters
    for g=1:length(kappa1_grid)

        % set param
        switch iParam
            case 1
                kappa1 = kappa1_grid(g);
            case 2
                kappa2 = kappa2_grid(g);
            case 3
                a_f = a_f_grid(g);
            case 4
                a_b = a_b_grid(g);
        end

        A = [0 1 0 0; -kappa1^2 -f1 a_f 0; 0 0 0 1; a_b 0 -kappa2^2 -f2
           ];

        % integrate diff. eq.
        x = zeros(4,length(t));
        for i=2:length(t)
            x(:,i) = x(:,i-1) + dt*(A*x(:,i-1)+C*u(i-1));
        end

        vE(g) = 1 - var(x_condition_2(:)-x(:))/var(x_condition_2(:));

    end

    [vE_maxParam(iParam), index_maxParam(iParam)] = max(vE);

    % reset parameter values
    switch iParam
        case 1
            kappa1=80;
        case 2
            kappa2=50;
        case 3
            a_f = 3000;
        case 4
            a_b = 1000;
    end
end

[vE_max, iminParam] = max(vE_maxParam);

fprintf("The model with a varied %s parameter explains the data best (%f
   %% explained variance)\n", varParams(iminParam), vE_max*100);
```

```
fprintf("the index in the param grid is: %i\n",index_maxParam(iminParam)
    );

fprintf("the optimal kappa1=%f\n",kappa1_grid(index_maxParam(iminParam))
    );
```

The model with a different κ_1 explains the data best. The optimal value is $\kappa_1 = 100$ (100% explained variance)

c

$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$