Biomedical Imaging Exercise – Week 10

Name: Alkinoos Sarioglou

Student ID: 20-947-743

1. Task 2.1

a) The differential equations for the concentrations of extracellular $c_e(t)$ and metabolized $^{18}F - FDG$ ($c_m(t)$) can be written as:

$$\frac{dc_e(t)}{dt} = k_1 \cdot c_p(t) - (k_2 + k_3) \cdot c_e(t)$$

$$\frac{dc_m(t)}{dt} = k_3 \cdot c_e(t)$$

with $c_p(t) = \delta(t)$, i.e. the delta dirac function.

The derived equations are the kinetic equations for the tracer and they express the impulse responses of the system's compartments $c_e(t)$ and $c_m(t)$.

2. Task 2.2

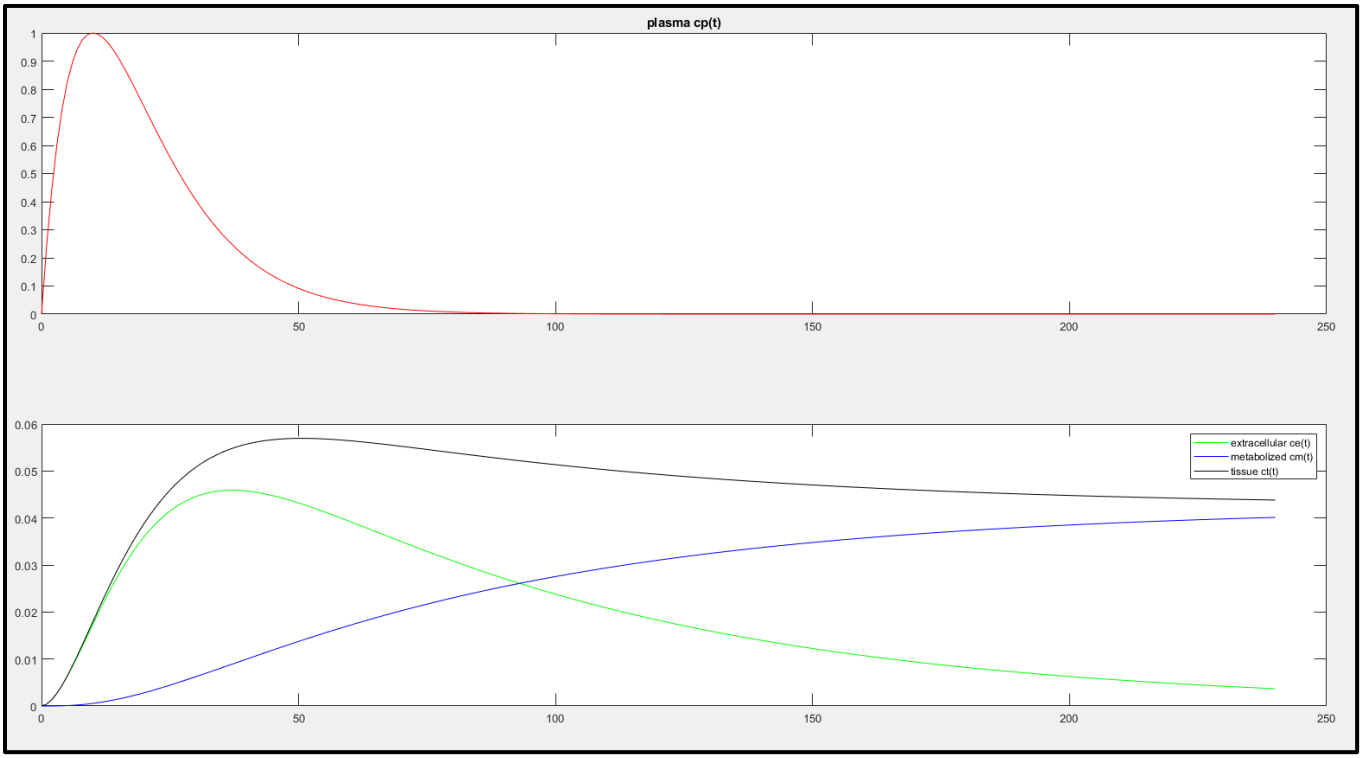
a) The differential equations derived before are implemented on MATLAB by using the "dsolve" function as following:

```
% ------------------------------------------------------------------------
% Set rate constants of kinetic model
% ------------------------------------------------------------------------
k(1) = 0.1;                                             %  [1/min]
k(2) = 0.3;                                             %  [1/min]
k(3) = 0.5;                                             %  [1/min]

% ------------------------------------------------------------------------
% Convert from [1/min] to [1/sec]
% ------------------------------------------------------------------------
k(:) = k(:)/60;                                         % [1/min -> 1/sec]

% ------------------------------------------------------------------------
% Task 2.1 Set up first-order differential equations to obtain impulse
% response function h(t) (see help dsolve)
% ------------------------------------------------------------------------
syms k1 k2 k3 ce(t) cm(t)

h = dsolve(diff(ce) == k1*dirac(t)- (k2+k3)*ce, diff(cm) == k3*ce, ce(0) == k1, cm(0) == 0);

% ------------------------------------------------------------------------
% Extract impulse response functions h_ce(t) and h_cm(t) from h(t)
% ------------------------------------------------------------------------
h_ce  = matlabFunction(h.ce);
h_cm  = matlabFunction(h.cm);
k1  = k(1);
k2  = k(2);
k3  = k(3);
```

b) The extracellular concentration $c_e(t)$ and the metabolized concentration $c_m(t)$ can be extracted by convolving the respective impulse responses with the blood plasma concentration $c_p(t)$. This is achieved in code by implementing:

```
% ------------------------------------------------------------------------
% Set time span for experiment
% ------------------------------------------------------------------------
t   = 0:240;                                        % [sec]

% ------------------------------------------------------------------------
% Set up input blood plasma function cp(t) using gamma variate pdf
% ------------------------------------------------------------------------
cp  = gampdf(t,2,10);                               % [ml/g]
cp  = cp/max(cp(:));

% ------------------------------------------------------------------------
% Task 2.2. Convolve cp(t) with impulse response h(t) to obtain
% extracellular concentration ce(t) and metabolized concentration cm(t)
% ------------------------------------------------------------------------

% ce(t)
ce = filter(cp,1,h_ce(k1,k2,k3,t));

% cm(t)
cm = filter(cp,1,h_cm(k1,k2,k3,t));

% ct(t)
ct = ce+cm;

% ------------------------------------------------------------------------
% Display cp(t), ce(t) and cm(t)
% ------------------------------------------------------------------------
figure(1); subplot(2,1,1); plot(t,cp,'red'); title('plasma cp(t)');
figure(1); subplot(2,1,2); plot(t,ce,'green'); hold on;
figure(1); subplot(2,1,2); plot(t,cm,'blue'); hold on;
figure(1); subplot(2,1,2); plot(t,ct,'black');
legend('extracellular ce(t)','metabolized cm(t)','tissue ct(t)');
```

c) For the specified constant values k_1, k_2, k_3, the concentration-time curves of $c_e(t)$ and $c_m(t)$ look like the following:



From the produced graphs it can be seen that $c_e(t)$ rises rapidly when $c_p(t)$ is increased with a rate of increase $k_1 = 0.1\ {}^{1}/_{min}$. Then, there is an exchange of tracer concentrations between the plasma and the extracellular space with rates $k_1 = 0.1\ {}^{1}/_{min}$ (from plasma to extracellular space) and $k_2 = 0.3\ {}^{1}/_{min}$ (from the extracellular space to plasma), until $c_p(t)$ reaches zero.

At the same time, the metabolized concentration $c_m(t)$ increases with a rate of $k_3 = 0.5\ {}^{1}/_{min}$ and stabilizes when the extracellular-space concentration $c_e(t)$ reduces to zero.

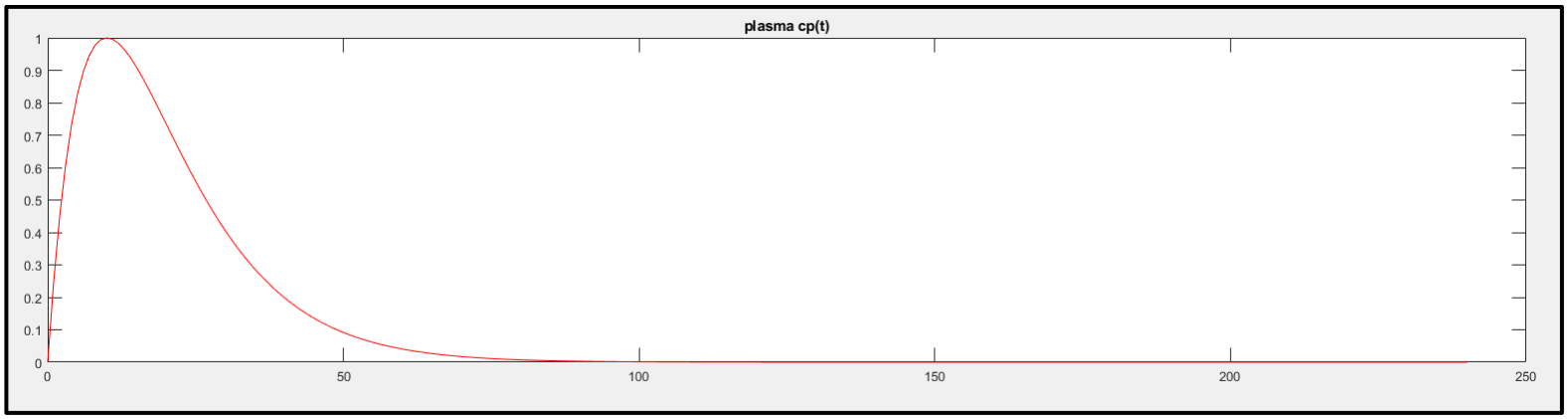
d) For $k_4 = k_{off}$ with $k_{off} = [L] \cdot [R]\ (1)$, which is the rate at which ligands L and cell receptors R are produced to be free. In the uptake of ${}^{18}F - FDG$, the tracer is converted to ${}^{18}F - FDG - 6 - P$. However, this is not converted back to ${}^{18}F - FDG$ in a high rate because the ${}^{18}F - FDG - 6 - P$ formed cannot move out of the cell before radioactive decay happens and gets trapped in cells. Therefore, $[L] \cong 0$, which means from (1) that $k_{off} = k_4 \cong 0$, which is a good assumption for this case.

3. Task 2.3

a) The blood plasma concentration-time curve $c_p(t)$ corresponds to a gamma probability density function with a shape parameter α=2 and an inverse scale parameter β=10:

```
% ------------------------------------------------------------------------
% Set up input blood plasma function cp(t) using gamma variate pdf
% ------------------------------------------------------------------------
cp   = gampdf(t,2,10);                                  % [ml/g]
cp   = cp/max(cp(:));
```

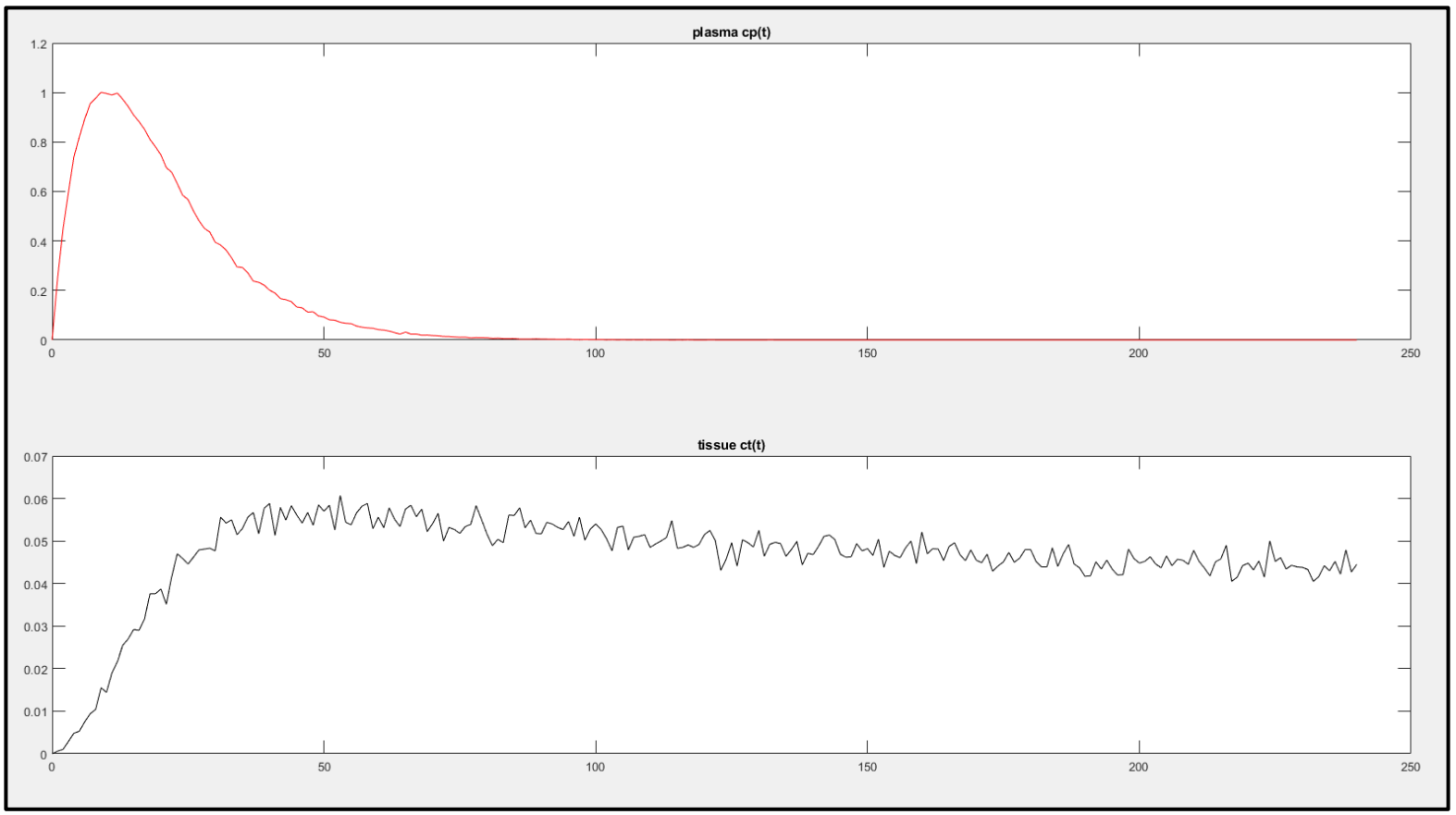
The concentration-time curve $c_p(t)$ looks like the following:



b) It is given that $SNR = \sqrt{\bar{N}}$, therefore for $SNR = 100$, then $\bar{N} = 100^2 = 10^4$ or more generally $\bar{N} = N0 = SNR^2$. Then, the photon count is extracted from the concentrations, Poisson noise is added and the noisy concentrations are derived again by executing the following code:

```
for i = 1:10

    % ------------------------------------------------------------------
    % Task 2.3. Convert concentrations into counts N by assuming
    % a peak SNR of blood plasma activity of 100
    % ------------------------------------------------------------------
    N0      = SNR^2;
    Ncp     = N0*cp;
    Nct     = N0*ct;

    % ------------------------------------------------------------------
    % Task 2.3. Add noise to cp(t) and ct(t) = ce(t) + cm(t)
    % ------------------------------------------------------------------
    Ncp_noise = poissrnd(Ncp,size(Ncp));
    Nct_noise = poissrnd(Nct,size(Nct));

    % ------------------------------------------------------------------
    % Task 2.3.Convert from counts N back to concentrations
    % ------------------------------------------------------------------
    cp_noise  = Ncp_noise/N0;
    ct_noise  = Nct_noise/N0;

    % ------------------------------------------------------------------
    % Display noisy concentration-time curves
    % ------------------------------------------------------------------
    figure(2); subplot(2,1,1); plot(t,cp_noise,'red');   title('plasma cp(t)'); hold on;
    figure(2); subplot(2,1,2); plot(t,ct_noise,'black'); title('tissue ct(t)'); hold on;
```

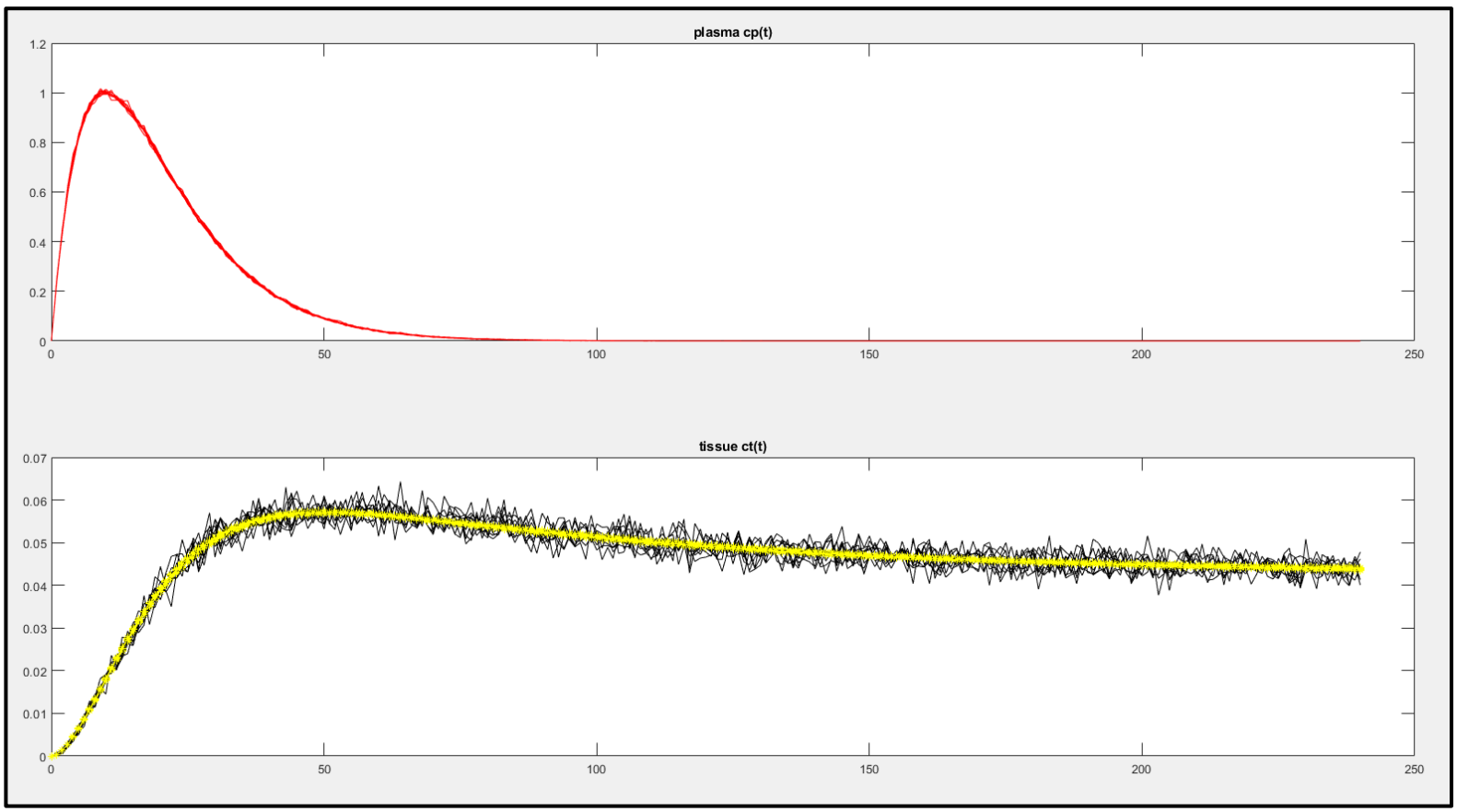
The resulting graphs demonstrate significant amount of Poisson noise both for the plasma concentration $c_p(t)$ and the tissue concentration $c_t(t)$:



c) The model function of the fit function is extracted by convolving the noisy $c_p(t)$ with the tissue impulse response h_t, which is a combination of the extracellular-space and the metabolized impulse responses h_ce and h_cm respectively. Then, the values of k_1, k_2, k_3 are extracted for 10 repetitions and the noisy $c_t(t)$ is plotted with these values. The above operations are executed by the following code:

```
% ------------------------------------------------------------------
% Fit noisy data input cp_noise, ct_noise using Matlab's nlinfit
% to obtain k1, k2, k3
% ------------------------------------------------------------------

% ------------------------------------------------------------------
% Task 2.3 Define fit function using h_t(t) = h_ce(t) + h_cm(t)
% ct_noise(t) = int(cp_noise(tau)*(h_t(t-tau))dtau
% use filter() to perform the convolution
% ------------------------------------------------------------------
fun = @(k,t)filter(cp_noise, 1, h_ce(k(1),k(2),k(3),t) + h_cm(k(1),k(2),k(3),t));

k0 = [0.01 0.01 0.01];                          % Starting values for fit

k_fit  = nlinfit(t,ct_noise,fun,k0);

% ------------------------------------------------------------------
% Display fitted concentration ct_fit(t) and estimated rates k1 k2 k3
% ------------------------------------------------------------------
figure(2); subplot(2,1,2); plot(t,fun(k_fit,t),'yellow*-');

k_result(:,i) = k_fit(:)*60;                    % [1/sec -> 1/min]

fprintf('Estimated rate constants [k1 k2 k3]: %f %f %f\n',k_result(:,i));
```

The resulting graphs look like the following:



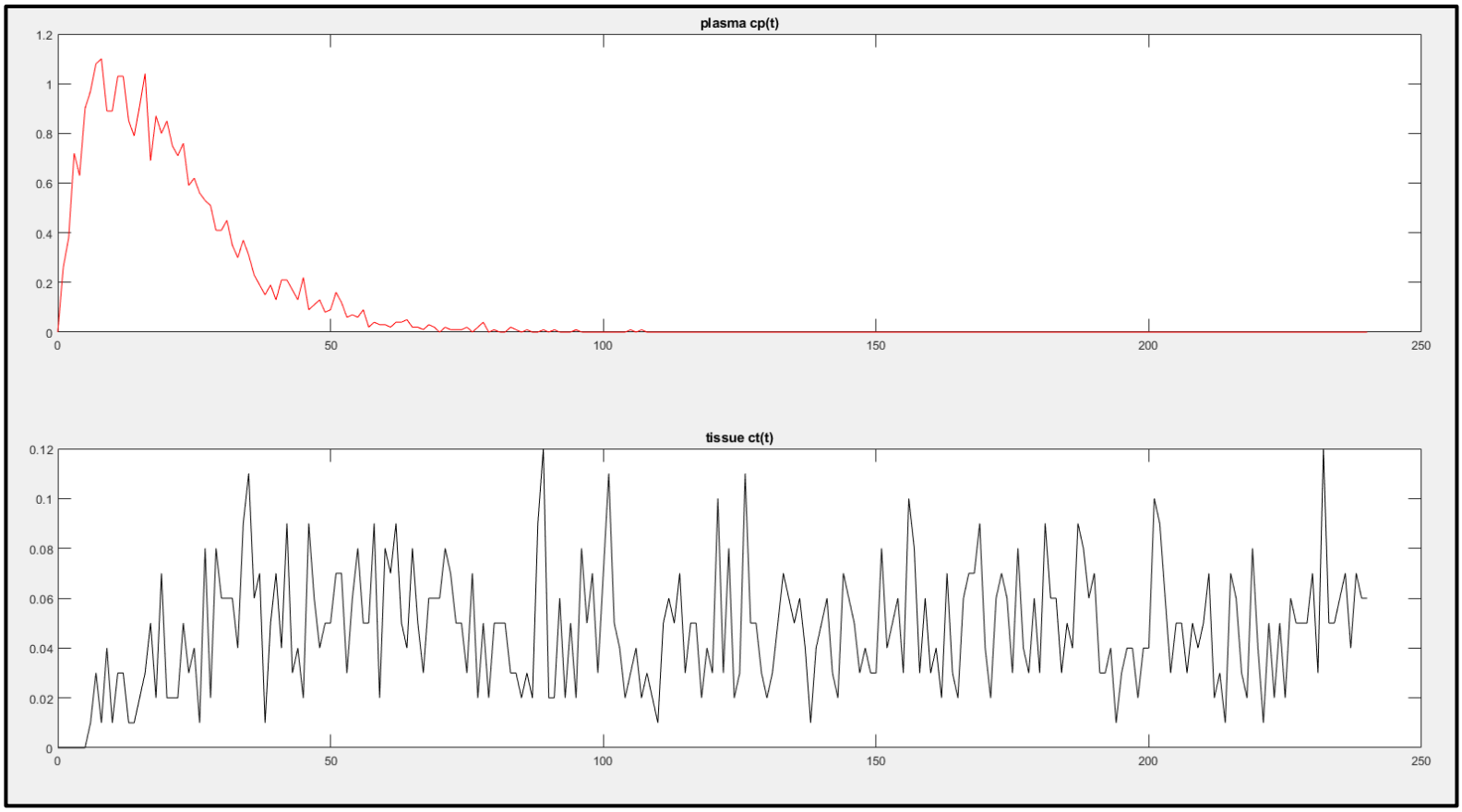
d) The mean and the standard deviation values of the constants are calculated by the following functions:

```
% ------------------------------------------------------------------------
% Task 2.3. Display mean +/- standard deviation of k1, k2, k3
% ------------------------------------------------------------------------
fprintf('Mean  [k1 k2 k3]: %f %f %f\n',mean(k_result,2));
fprintf('StDev [k1 k2 k3]: %f %f %f\n',std(k_result,1,2));
```

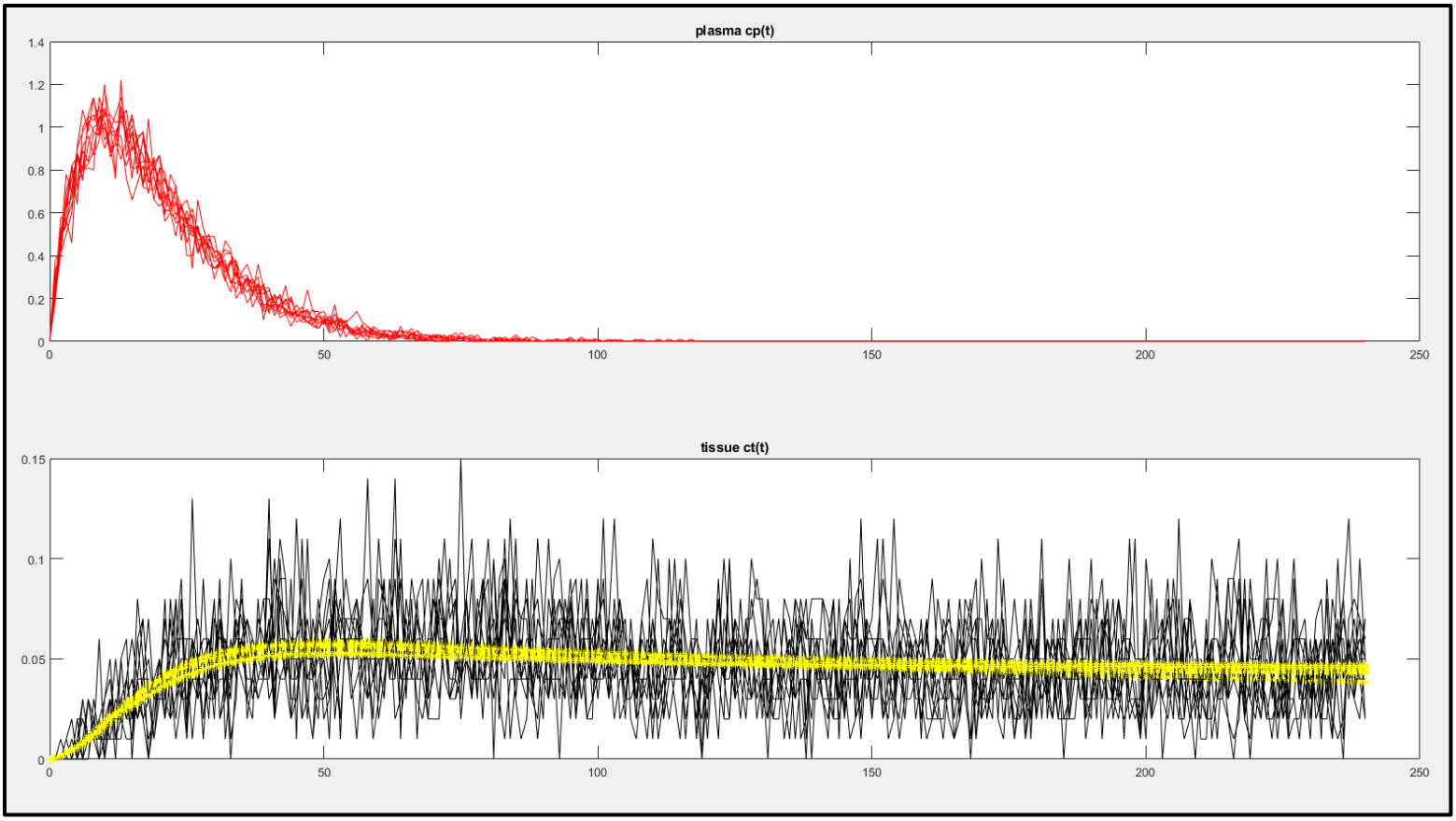
For $SNR = 100$, the values printed on the terminal are the following:

```
Estimated rate constants [k1 k2 k3]: 0.099594 0.286792 0.470203
Estimated rate constants [k1 k2 k3]: 0.101647 0.316087 0.514578
Estimated rate constants [k1 k2 k3]: 0.101820 0.327566 0.524954
Estimated rate constants [k1 k2 k3]: 0.101631 0.340679 0.563726
Estimated rate constants [k1 k2 k3]: 0.101349 0.322073 0.527650
Estimated rate constants [k1 k2 k3]: 0.099186 0.281541 0.474725
Estimated rate constants [k1 k2 k3]: 0.100694 0.315348 0.539098
Estimated rate constants [k1 k2 k3]: 0.098307 0.263003 0.433376
Estimated rate constants [k1 k2 k3]: 0.100809 0.315175 0.527599
Estimated rate constants [k1 k2 k3]: 0.101269 0.320085 0.526960
Mean  [k1 k2 k3]: 0.100631 0.308835 0.510287
StDev [k1 k2 k3]: 0.001139 0.022657 0.036858
```

e) For $SNR = 10$, the graphs of $c_p(t)$ and $c_t(t)$ look like the following:



Furthermore, the fit function after 10 repetitions is illustrated below:



Also, the output mean and standard deviation values of the constants k_1, k_2, k_3 are:

```
Estimated rate constants [k1 k2 k3]: 0.098643 0.353675 0.585324
Estimated rate constants [k1 k2 k3]: 0.116193 0.830402 1.101715
Estimated rate constants [k1 k2 k3]: 0.108940 0.503652 0.724255
Estimated rate constants [k1 k2 k3]: 0.090387 0.136958 0.306503
Estimated rate constants [k1 k2 k3]: 0.120565 0.988847 1.274081
Estimated rate constants [k1 k2 k3]: 0.109564 0.311680 0.312334
Estimated rate constants [k1 k2 k3]: 0.093749 0.432521 0.972680
Estimated rate constants [k1 k2 k3]: 0.096676 0.424571 0.854947
Estimated rate constants [k1 k2 k3]: 0.107458 0.426975 0.491612
Estimated rate constants [k1 k2 k3]: 0.081812 0.063081 -0.041523
Mean   [k1 k2 k3]: 0.102399 0.447236 0.658193
StDev  [k1 k2 k3]: 0.011532 0.267481 0.385224
```

It is obvious that the output graphs are much noisier than before. However, the estimated $c_t(t)$ curve (yellow curve) follows the correct trend with relative accuracy. The mean values are very different than with $SNR = 100$, especially k_2 and k_3. The standard deviation is also higher, demonstrating that the low SNR has influenced the precision of the constant estimates significantly. Additionally, the SNR influences the plasma concentration $c_p(t)$ and as a result the tissue concentration $c_t(t)$ as well. Therefore, the SNR of the input PET data needs to be significantly high (at least 100), in order to get accurate and precise estimations of the fitted rate constants k_1, k_2, k_3.