Table of Contents

Baseline Models

Two baseline model for comparison. The first method uses robust fit (linear regression) to remove the back propogation components. The second method uses foopsi first, and then remove bAP in spike domain.

```
set(groot,'defaultLineLineWidth',0.8)
thr_val = 98;
```

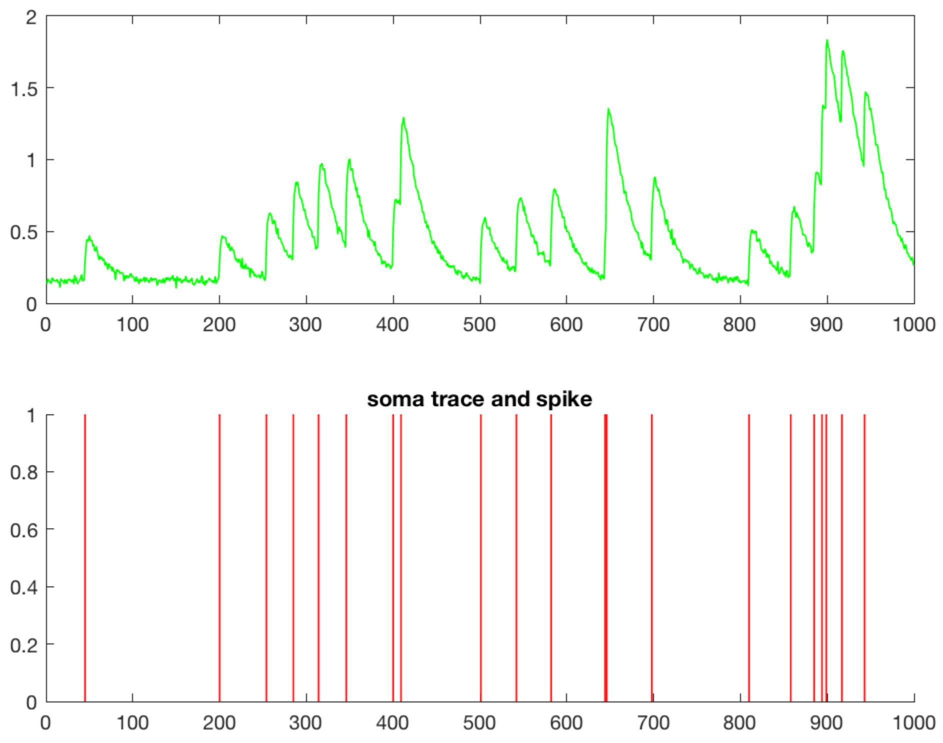
Loading the data

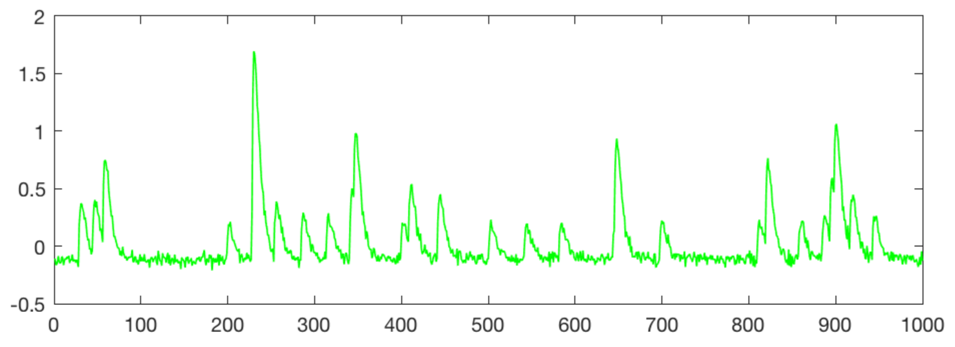
```
load('mydata.mat');
x_data = data.x_data;
y_data = data.y_data;

spikes = horzcat(x_data.s_s, x_data.s_d);
traces = horzcat(y_data.f_s, y_data.f_d);
fprintf('Number of soma traces: %d\n', size(x_data.s_s, 2))
fprintf('Number of spine traces: %d\n', size(x_data.s_d, 2))

% take the soma trace
soma_select = 2;
spine_select = 7;
length = 1000;

soma_trace = traces(1:length, soma_select);
soma_spike = spikes(1:length, soma_select);
spine_trace = traces(1:length, spine_select);
spine_spike = spikes(1:length, spine_select);
```

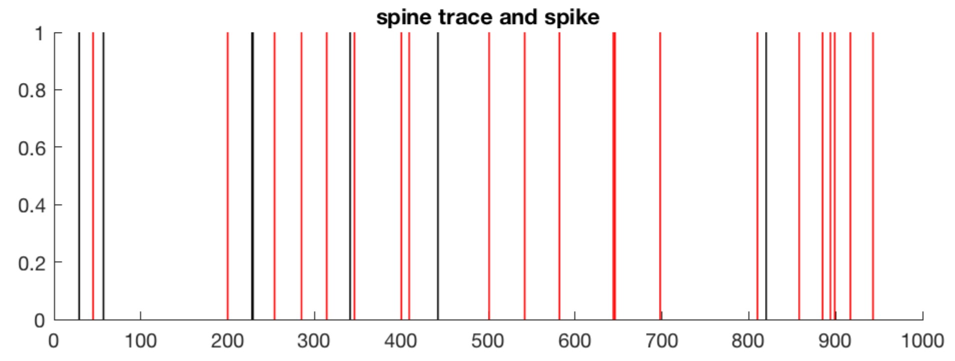
```
Number of soma traces: 6
Number of spine traces: 10
```

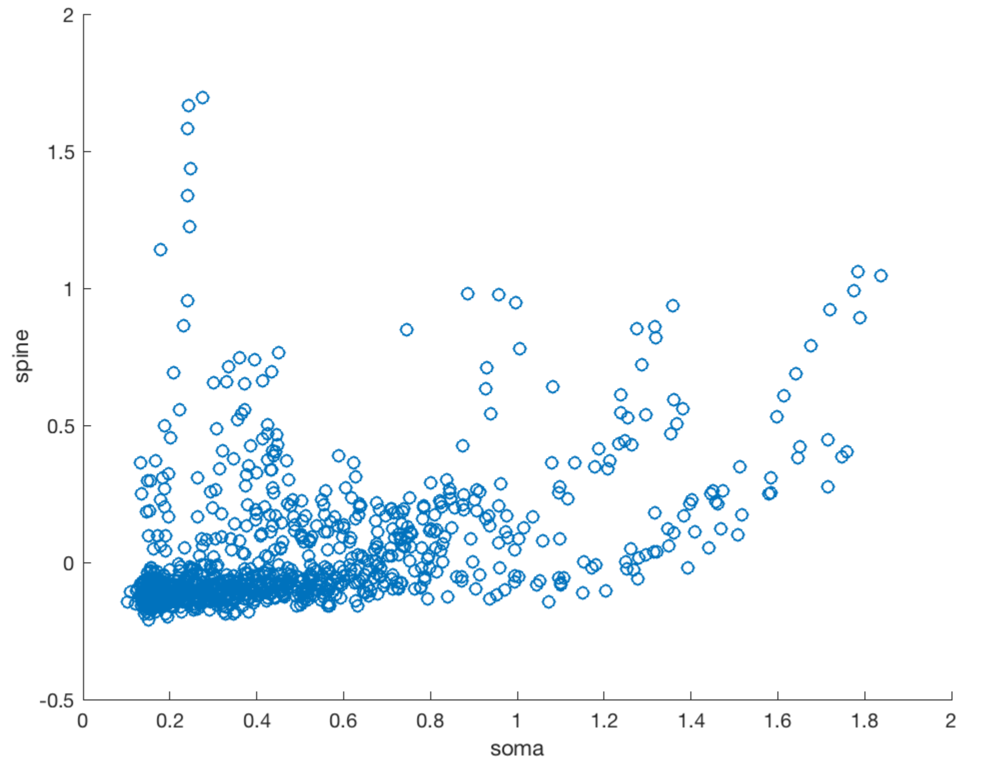
```
close all;
% plot the traces
figure(1)
subplot(211)
plot(soma_trace, 'g');
subplot(212)
plot_spike(soma_spike, 'r')
title('soma trace and spike')

% plot the traces
```

```
figure(2)
subplot(211)
plot(spine_trace, 'g');
subplot(212)
plot_spike(spine_spike, 'k'); hold on;
plot_spike(soma_spike, 'r'); hold off;
title('spine trace and spike')

%
figure(3)
scatter(soma_trace, spine_trace);
xlabel('soma');
ylabel('spine');
```



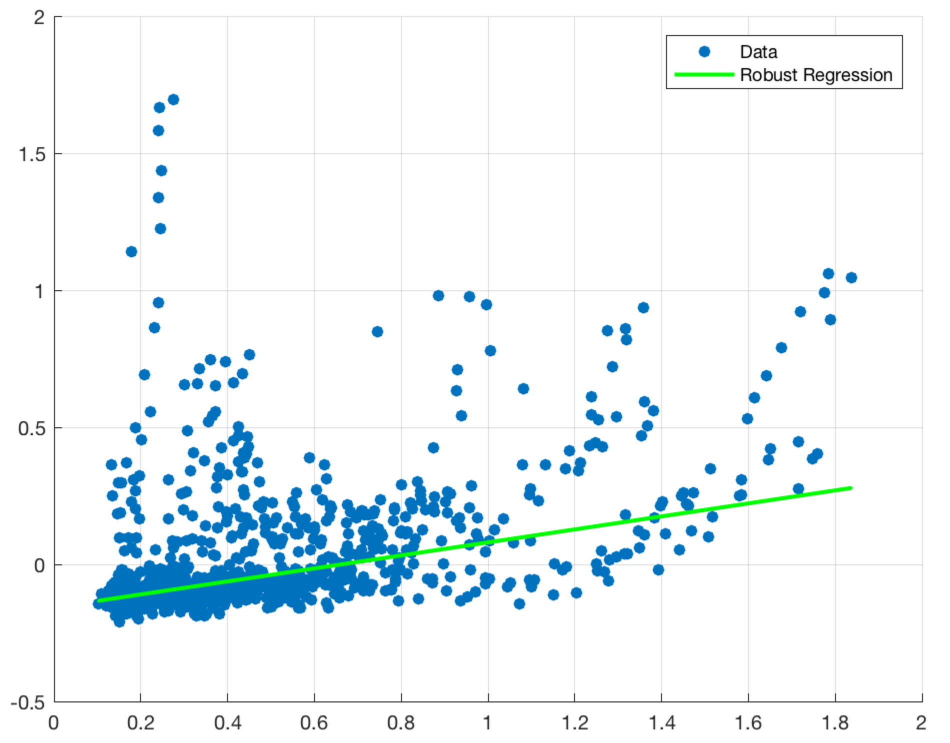

2
1.5
1
0.5
0
-0.5
0
100
200
300
400
500
600
700
800
900
1000

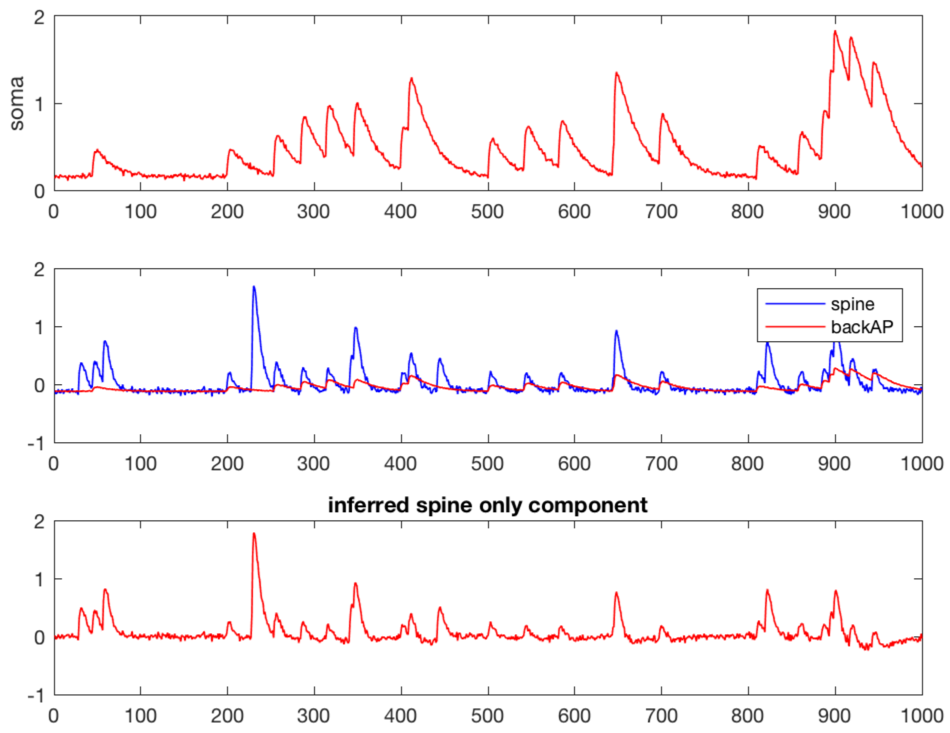

spine trace and spike
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000


2
1.5
1
0.5
0
-0.5
spine
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
soma

Robust Regression

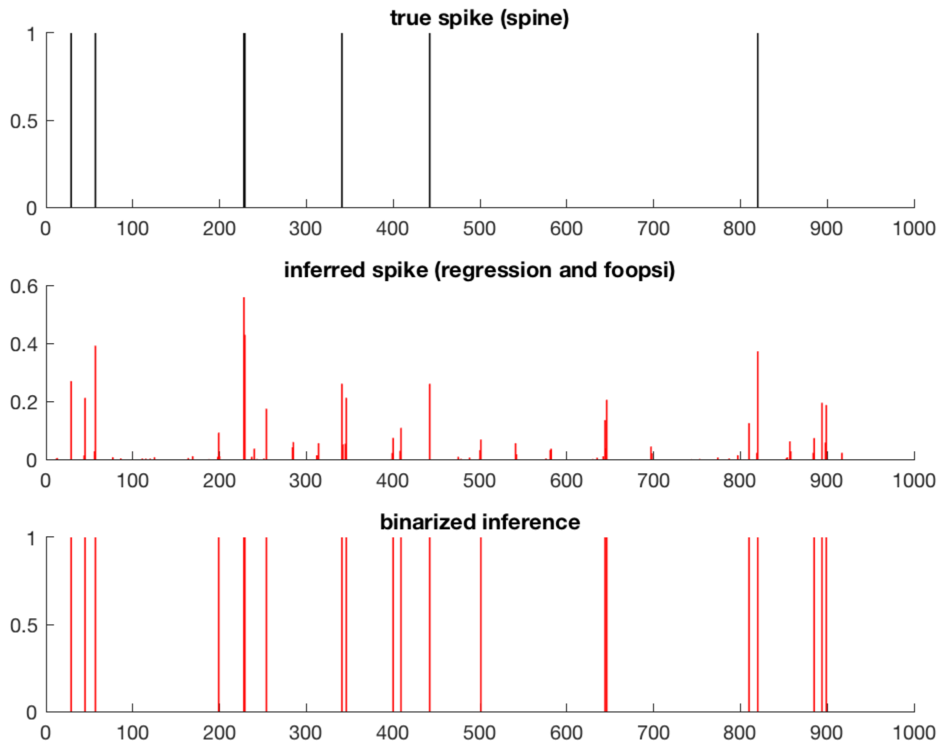
```
brob = robustfit(soma_trace, spine_trace);

% scatter plot of the data with the fits
figure(4)
scatter(soma_trace, spine_trace, 'filled'); grid on; hold on
plot(soma_trace, brob(1) + brob(2)*soma_trace, 'g', 'LineWidth', 2)
legend('Data', 'Robust Regression')

% plot the traces
figure(5)
subplot(311)
plot(soma_trace, 'r')
ylabel('soma')

subplot(312)
plot(spine_trace, 'b'); hold on;
plot(brob(1) + brob(2)*soma_trace, 'r');
hold off;
legend('spine', 'backAP')

spine_only_comp = spine_trace - (brob(1) + brob(2)*soma_trace);
subplot(313)
plot(spine_only_comp, 'r');
title('inferred spine only component')
```


Data
Robust Regression


soma
spine
backAP
inferred spine only component

As we can see, because of the dendrite (soma) and spine has different delay. The scatter plot is not strickly two components as in Chen, et.al, Nature (2013). This method doesn't take into account of different delays or any kind of nonlinearities.

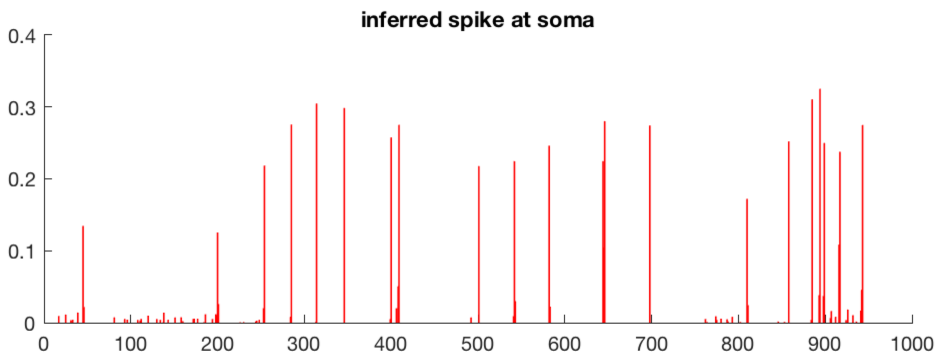
```
% spike inference using foopsi
addpath('constrained-foopsi-master')
[c,b,c1,g,sn,sp] = constrained_foopsi(spine_only_comp);

figure(6)
subplot(311)
plot_spike(spine_spike, 'k')
title('true spike (spine)')

subplot(312)
plot_spike(sp, 'r');
title('inferred spike (regression and foopsi)')


% threholding it
subplot(313)
thr_per = prctile(sp, thr_val);
sp_binary = double((sp >= thr_per));
plot_spike(sp_binary, 'r');
title('binarized inference')
```

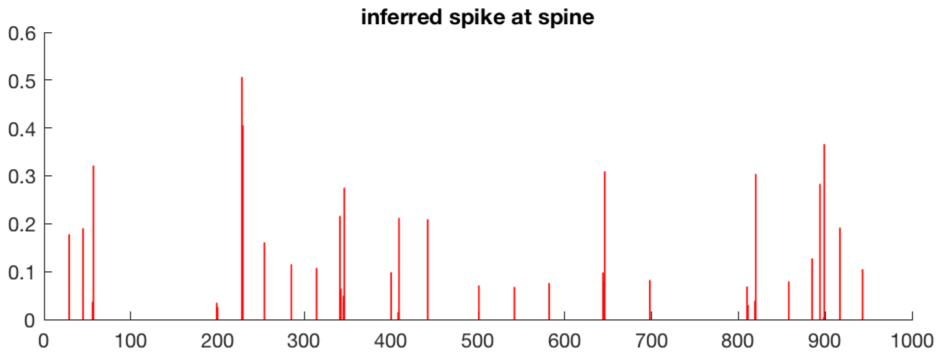


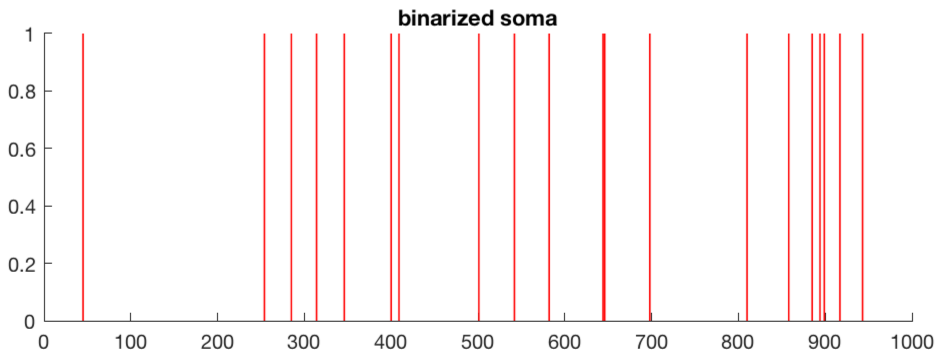
As we can see, there are many more inferred spikes than the true spikes.

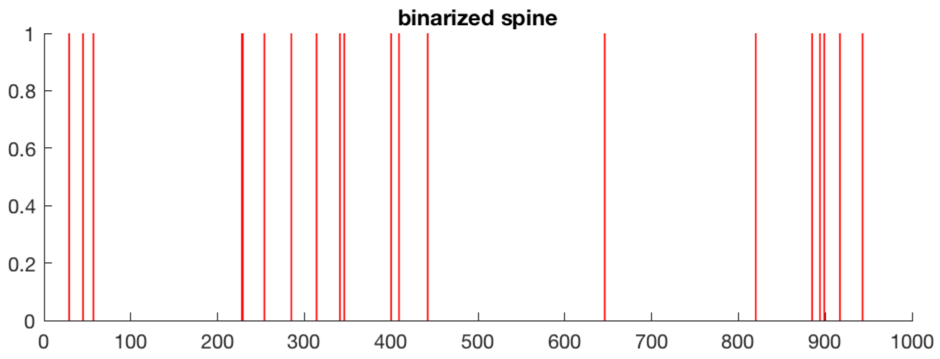
Two-step Foopsi

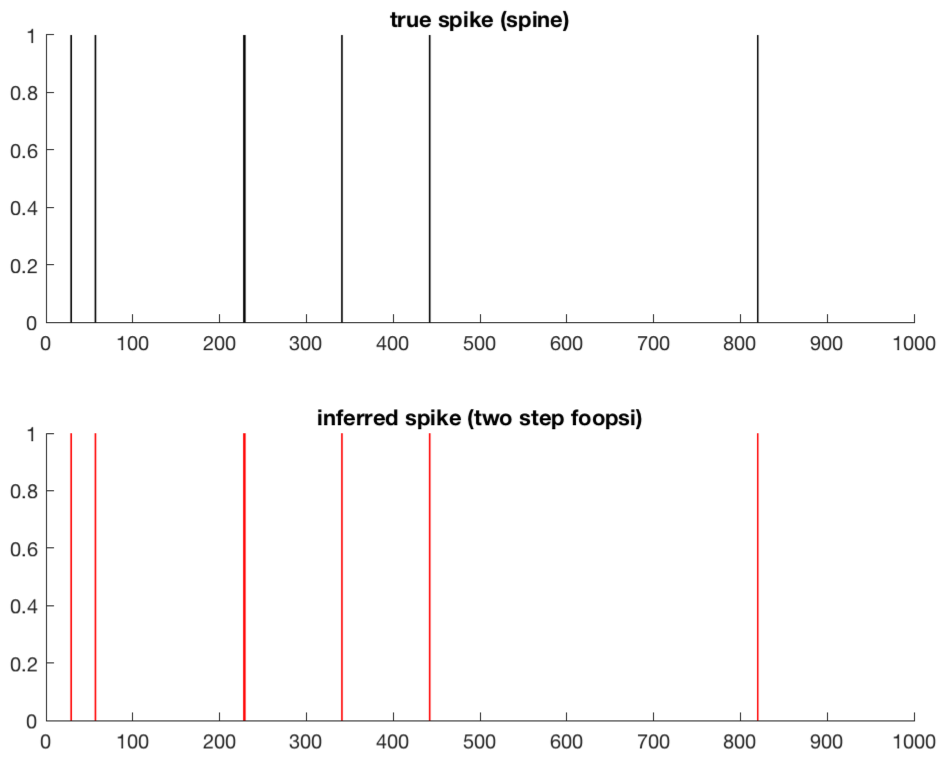
```
[c_soma,b_soma,c1_soma,g_soma,sn_soma,sp_soma] =
 constrained_foopsi(soma_trace);
[c_spine,b_spine,c1_spine,g_spine,sn_spine,sp_spine] =
 constrained_foopsi(spine_trace);

figure(7)
subplot(211)
plot_spike(sp_soma, 'r')
title('inferred spike at soma')

subplot(212)
plot_spike(sp_spine, 'r');
title('inferred spike at spine')

% threholding it
figure(8)
thr_per = prctile(sp_soma, thr_val);
sp_binary_soma = double((sp_soma >= thr_per));
subplot(211)
plot_spike(sp_binary_soma, 'r');
title('binarized soma')



thr_per = prctile(sp_spine, thr_val);
sp_binary_spine = double((sp_spine >= thr_per));
subplot(212)
plot_spike(sp_binary_spine, 'r');
title('binarized spine')

% substract
sp_spine_ind = sp_binary_spine - sp_binary_soma;
sp_spine_ind(sp_spine_ind<0) = 0;

figure(9)
subplot(211)
plot_spike(spine_spike, 'k')
title('true spike (spine)')

subplot(212)
plot_spike(sp_spine_ind, 'r');
title('inferred spike (two step foopsi)')
```


inferred spike at soma


inferred spike at spine


binarized soma


binarized spine



Two step foopsi is working pretty good when the data is quite linear and low noise.

Published with MATLAB® R2016a