HOMEWORK #6 SOLUTIONS

1. (a) The observed signal $x(n)$ and corrupted signal $s(n)$ are shown in Fig. 1.

   (b) To solve for $\hat{A}(z)$ from observed data $\{x(-P), x(-P+1), \dots, x(M-1)\}$, we create the P^{th}-order AR model

$$\underbrace{\begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(M-1) \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} x(-1) & x(-2) & \cdots & x(-P) \\ x(0) & x(-1) & \cdots & x(-P+1) \\ \vdots & \vdots & & \vdots \\ x(M-2) & x(M-1) & \cdots & x(M-P-1) \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{pmatrix}}_{\mathbf{a}} + \underbrace{\begin{pmatrix} e(0) \\ e(1) \\ \vdots \\ e(M-1) \end{pmatrix}}_{\mathbf{e}_x}$$

   which implies that prediction error can be written

$$\mathbf{e}_x \;=\; \mathbf{x} - \mathbf{X}\mathbf{a}$$

   and thus sum-squared prediction error can be written

$$\|\mathbf{e}_x\|^2 \;=\; \|\mathbf{x} - \mathbf{X}\mathbf{a}\|^2 \;=\; (\mathbf{x} - \mathbf{X}\mathbf{a})^t(\mathbf{x} - \mathbf{X}\mathbf{a}) \;=\; \mathbf{x}^t\mathbf{x} - 2\mathbf{a}^t\mathbf{X}^t\mathbf{x} + \mathbf{a}^t\mathbf{X}^t\mathbf{X}\mathbf{a}.$$

   Finding $\mathbf{a}$ which minimizes $\|\mathbf{e}_x\|^2$ can be accomplished by standard vector calculus, yielding least-squares estimate

$$\hat{\mathbf{a}} \;=\; (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{x}.$$

   The polynomial $\hat{A}(z) = \sum_{\ell=1}^{P} \hat{a}_\ell z^{-\ell}$ can be constructed from $\hat{\mathbf{a}} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_P)^t$. Simulating the data model specified in the homework, we find that

$$\begin{aligned} \hat{A}(z) &= 0.5439z^{-1} + 0.2003z^{-2} - 0.0068z^{-3} + 0.0599z^{-4} - 0.0336z^{-5}, \quad \text{where} \\ A(z) &= 3.1166z^{-1} - 3.8769z^{-2} + 2.2661z^{-3} - 0.5184z^{-4}. \end{aligned}$$

   Thus $\hat{A}(z)$ and $A(z)$ do not appear similar at all, which can be attributed to the fact that $x(n)$ is a very noisy version of $s(n)$.

   (c) We compute the prediction error sequence resulting from the model $\hat{A}(z)$ via

$$\mathbf{e}_x \;=\; \mathbf{x} - \mathbf{X}\hat{\mathbf{a}}.$$

   Sorting the elements in $\mathbf{e}_x$ and throwing away the largest 5%, we compute $\sigma_e = 0.2826$. Thresholding the prediction error sequence $e_x(n)$ at $3\sigma_e$ we estimate the corrupted indices as

$$\begin{aligned} \hat{\mathcal{N}_i} &= \{27, 30, 31, 37, 38, 39, 44, 48, 51, 52, 63, 64, 79, 80, 81\}, \quad \text{where} \\ \mathcal{N}_i &= \{6, 27, 30, 48, 51, 63, 75, 79, 80, 139\}. \end{aligned}$$

   The sets $\mathcal{N}_i$ and $\hat{\mathcal{N}_i}$ share various points, though a number of corrupted locations in $\mathcal{N}_i$ were not detected and a number of detections in $\hat{\mathcal{N}_i}$ were false alarms. See also Fig. 1. (Note: the plot starts the data index at $n = -P$ rather than at $n = 1$.)

(d) To estimate $s(n)$ at the though-to-be-corrupted locations $\hat{\mathcal{N}}_i$, we formulate the AR model in a different way:

$$\underbrace{\begin{pmatrix} e(0) \\ e(1) \\ \vdots \\ e(M-1) \end{pmatrix}}_{\mathbf{e}_s} = \underbrace{\begin{pmatrix} -\hat{a}_P & \cdots & -\hat{a}_1 & 1 & & & & \\ & -\hat{a}_P & \cdots & -\hat{a}_1 & 1 & & & \\ & & -\hat{a}_P & \cdots & -\hat{a}_1 & 1 & & \\ & & & \ddots & & \ddots & \ddots & \\ & & & & -\hat{a}_P & \cdots & -\hat{a}_1 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s(-P) \\ s(-P+1) \\ \vdots \\ s(M-1) \end{pmatrix}}_{\mathbf{s}}$$

The model can be easily partitioned into "known" and "unknown" components:

$$\mathbf{e}_s \;=\; \mathbf{A}_\mathrm{k}\mathbf{s}_\mathrm{k} + \mathbf{A}_\mathrm{u}\mathbf{s}_\mathrm{u}$$

where $\mathbf{s}_\mathrm{u}$ contains the elements of $\mathbf{s}$ whose indices are in the set $\hat{\mathcal{N}}_i$ and $\mathbf{s}_\mathrm{k}$ contains the remaining elements, and where $\mathbf{A}_\mathrm{k}$ and $\mathbf{A}_\mathrm{u}$ are formed from the corresponding columns of $\mathbf{A}$. The vector $\mathbf{s}_\mathrm{u}$ minimizing $\|\mathbf{e}_s\|^2$ can be found (via standard vector calculus) to be

$$\hat{\mathbf{s}}_\mathrm{u} \;=\; -\,(\mathbf{A}_\mathrm{u}^t\mathbf{A}_\mathrm{u})^{-1}\mathbf{A}_\mathrm{u}^t\mathbf{A}_\mathrm{k}\mathbf{s}_\mathrm{k}.$$

The sequence $\{\hat{s}(n)\}$ is obtained by correctly indexing the elements from $\hat{\mathbf{s}}_\mathrm{u}$ and $\mathbf{s}_\mathrm{k}$. (See Fig. 1 for a plot.) The restoration process has been successful in removing the large noise spikes, but the restored signal is not exactly equal to the noiseless signal.

(e) See Fig. 1.

2. (a) Using the sequence $\{\hat{s}(n)\}$ and the technique of 1(b), we get a better estimate of $A(z)$:

$$\begin{aligned} \breve{A}(z) &= 1.3866z^{-1} - 0.1771z^{-2} - 0.3652z^{-3} - 0.1619z^{-4} + 0.2520z^{-5}, \quad \text{where} \\ A(z) &= 3.1166z^{-1} - 3.8769z^{-2} + 2.2661z^{-3} - 0.5184z^{-4}. \end{aligned}$$

than we did using the sequence $\{x(n)\}$. This is because the worst corrupting noise in $\{x(n)\}$ has been removed in forming $\{\hat{s}(n)\}$.

(b) Estimating the corrupted indices using the technique in 1(c), we find

$$\begin{aligned} \breve{\mathcal{N}}_i &= \{6, 7, 27, 30, 38, 39, 40, 41, 48, 51, 63, 75, 79, 80, 81, 83, 140\} \quad \text{where} \\ \mathcal{N}_i &= \{6, 27, 30, 48, 51, 63, 75, 79, 80, 139\}. \end{aligned}$$

Clearly $\breve{\mathcal{N}}_i$ is a better estimate of $\mathcal{N}_i$ than is $\hat{\mathcal{N}}_i$, though the estimate is by no means perfect.

(c) The result of using $\breve{A}(z)$ and $\breve{\mathcal{N}}_i$ to estimate $\{s(n)\}$ is shown in Fig. 2. The estimate $\{\breve{s}(n)\}$ is quite good—definitely better than $\{\hat{s}(n)\}$—and the only substantial departure from $\{s(n)\}$ occurs in the second-to-last signal peak.

(d) See Fig. 2.

(e) The third and fourth estimations of $A(z)$ were:

$$\begin{aligned} \breve{A}_3(z) &= 2.0933z^{-1} - 1.3824z^{-2} + 0.1259z^{-3} + 0.0041z^{-4} + 0.1375z^{-5} \\ \breve{A}_4(z) &= 3.1174z^{-1} - 3.8666z^{-2} + 2.1847z^{-3} - 0.3982z^{-4} - 0.0494z^{-5}, \end{aligned}$$

and the latter is very close to the true $A(z)$. The corresponding estimations of $\mathcal{N}_i$ were:

$$\begin{aligned} \breve{\mathcal{N}}_{i,3} &= \{6, 27, 30, 48, 51, 63, 75, 79, 80, 139, 140\} \\ \breve{\mathcal{N}}_{i,4} &= \{6, 27, 30, 48, 51, 63, 75, 79, 80, 139\}, \end{aligned}$$

the last of which equals the true set of corruption locations $\mathcal{N}_i$! Figures 3 and 4 show the corresponding estimates of $\{s(n)\}$, which appear nearly perfect.

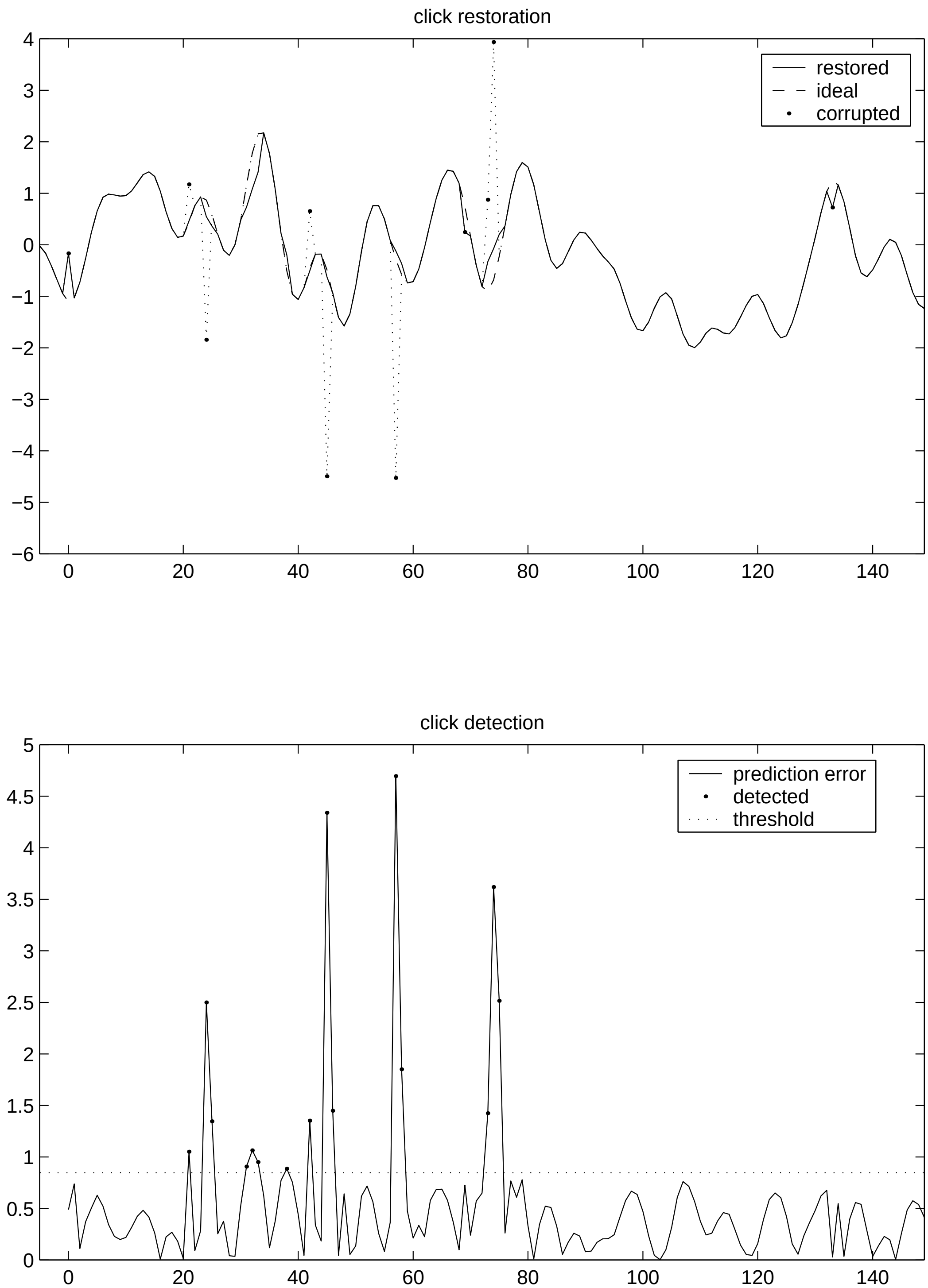

Figure 1: First round of click detection/restoration

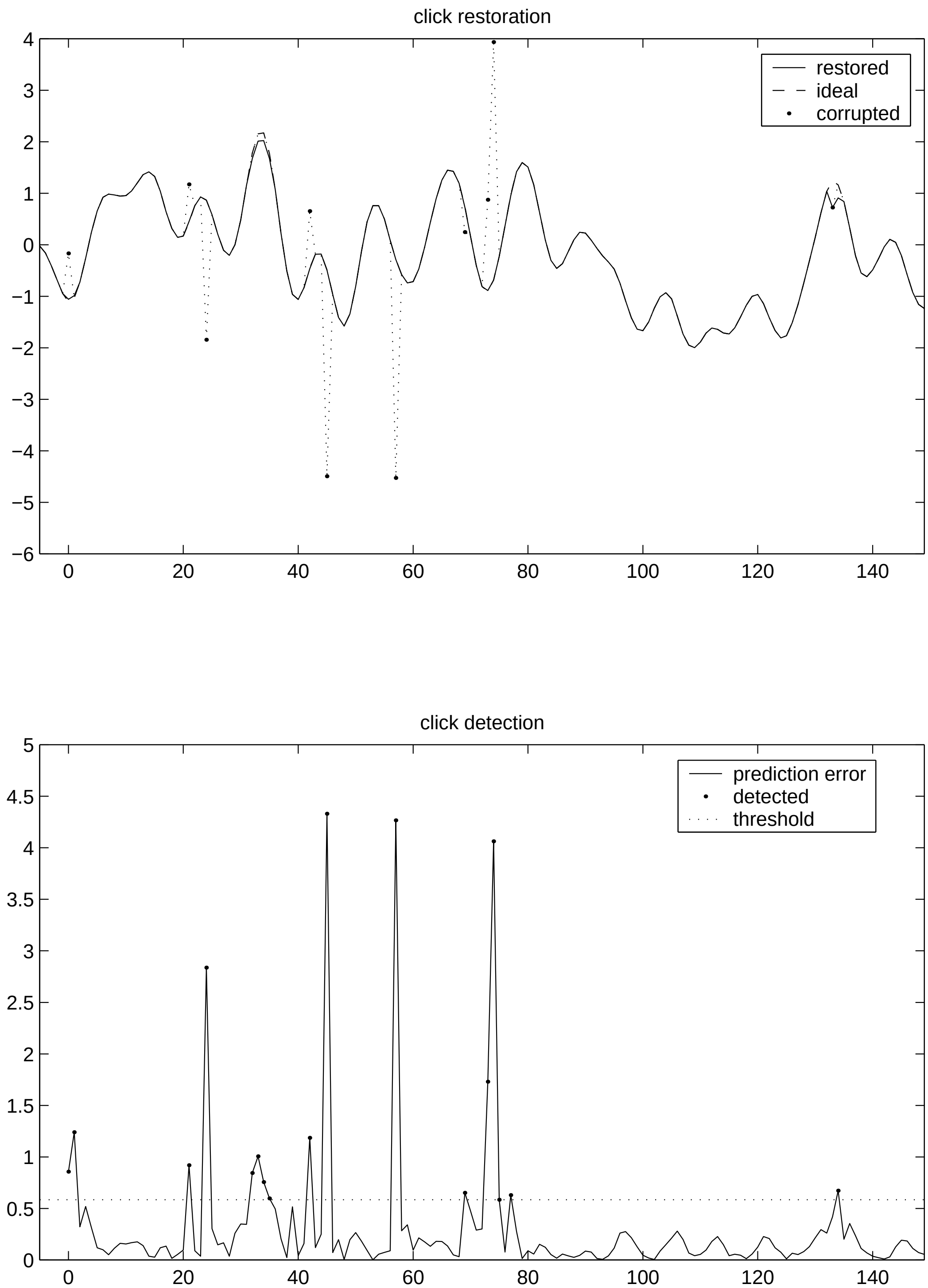

Figure 2: Second round of click detection/restoration

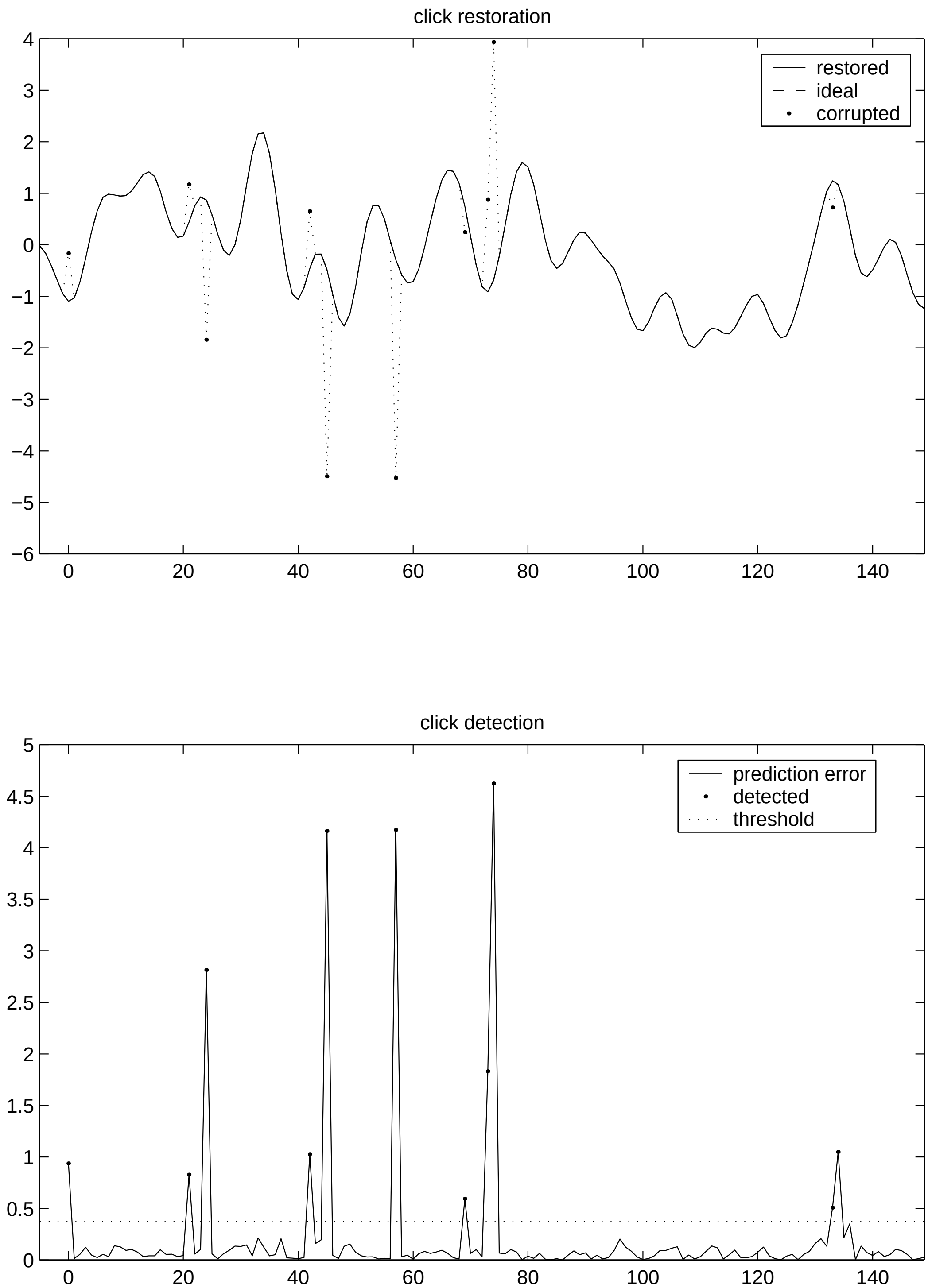

Figure 3: Third round of click detection/restoration

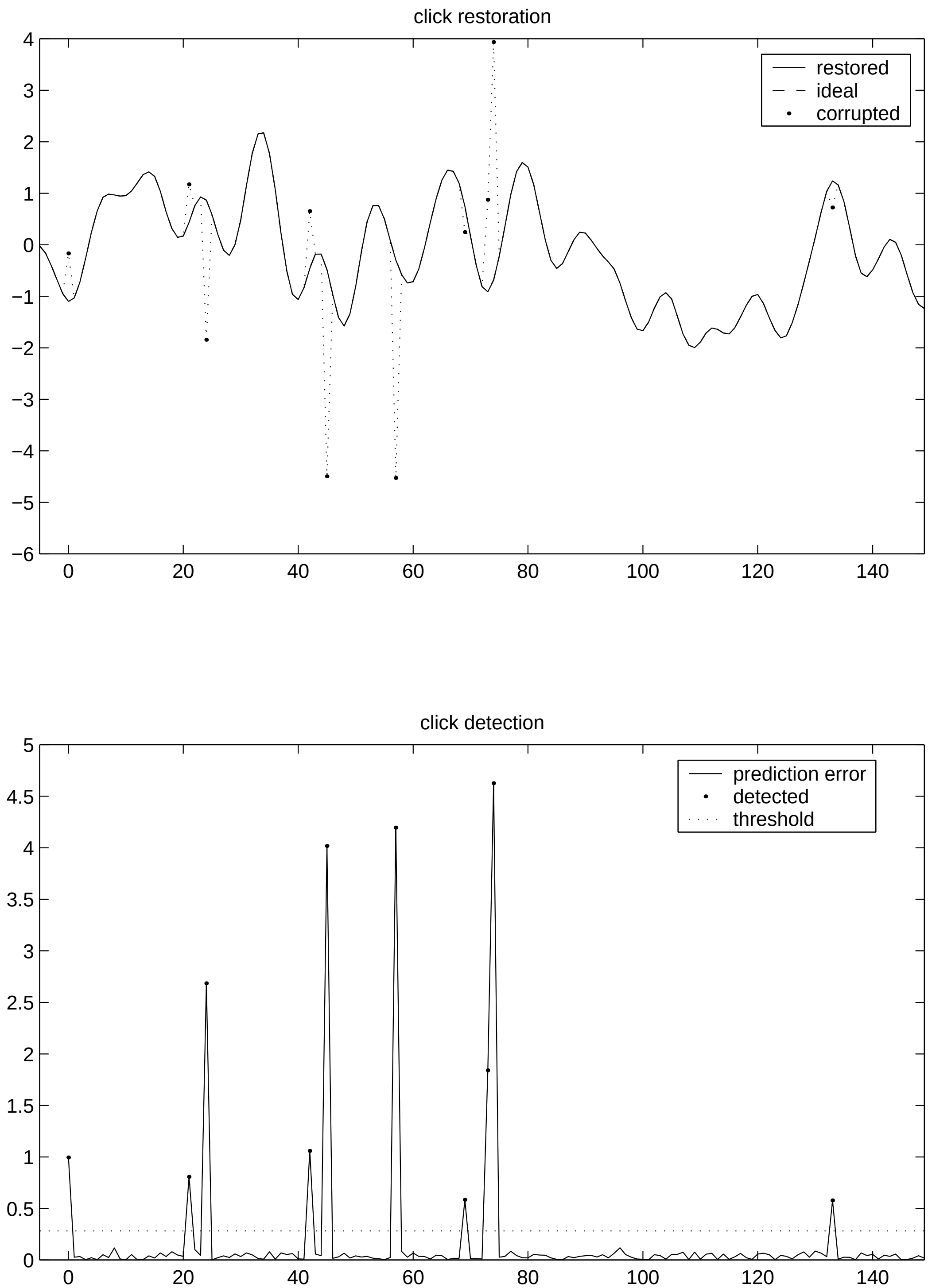

Figure 4: Fourth round of click detection/restoration

Matlab Code:

```
% LSAR detection/reconstruction for homework6

 % reset seed
 randn('state',0);

 % parameters
 M = 150; % block length
 P = 5; % AR model order
 outlier = 3; % used in error thresholding

 % create AR signal
%poles = [0.8,0.9].*exp(j*2*pi*[0.01,0.09]);
%tmp = poly([poles,conj(poles)]);  a_true = -tmp(2:5);
 a_true = [3.1166   -3.8769    2.2661    -0.5184];
%freqz(1,[1 -a_true])
 s = filter(1,[1 -a_true],randn(1,M+P));
 s = s/sqrt(var(s)); % unit variance

 % add noise bursts
%i = sort(P+ceil(M*rand(1,round(M/10)))); % random corruption indices
 i = [6    27    30    48    51    63    75    79    80    139   ];
 n = 2*randn(1,M+P); % noise process
 x = s;
 x(i) = s(i)+n(i); % corruption at indices i

 % estimate AR model
 xvec = x(P+[1:M]).';
 X = zeros(M,P); for p=1:P, X(:,p) = x(P-p+[1:M])'; end;
 a = X\xvec; % AR coefficient estimates (i.e., a = pinv(X)*xvec;)
 A = [zeros(M,P),eye(M)];
 aflip = -fliplr(a.'); for n=1:M, A(n,n+[0:P-1]) = aflip; end;
 e = xvec-X*a; % prediction error

 % detect corruption indices
 e_ord = sort(abs(e));
 sig_e = sqrt(var( e_ord(1:ceil(0.95*length(e_ord))) )); % remove outliers
 thresh = outlier*sig_e;
 ibad = find(abs(e)>thresh)+P;
 igood = setdiff([1:M+P],ibad);

 % interpolate
 sh = x;
 sh(ibad) = -A(:,ibad)\(A(:,igood)*x(igood).');

 figure(1);
 subplot(211)
   plot([-P:M-1],sh,'-',[-P:M-1],s,'--r',[i-P-1],x(i),'.g',[-P:M-1],x,':k');
   axe = axis; axis([-P,M-1,axe(3:4)]);
   title('click restoration');
   legend('restored','ideal','corrupted',0);
 subplot(212)
   plot([0:M-1],abs(e),'g',[ibad-P-1],abs(e(ibad-P)),'.m');
   hold on; plot([-P,M-1],thresh*[1,1],'k:'); hold off;
   axe = axis; axis([-P,M-1,axe(3:4)]);
   title('click detection');
   legend('prediction error','detected','threshold',0);
 orient tall;

 % iteratively re-estimate quantities
 for r=1:3,
   % re-estimate AR model
   shvec = sh(P+[1:M]).';
   Sh = zeros(M,P); for p=1:P, Sh(:,p) = sh(P-p+[1:M])'; end;
   a = Sh\shvec;  % AR coefficient estimates (i.e., a = pinv(Sh)*shvec;)
   aflip = -fliplr(a.'); for n=1:M, A(n,n+[0:P-1]) = aflip; end;

   % detect corruption indices
   e = xvec-Sh*a; % error
   e_ord = sort(abs(e));
   sig_e = sqrt(var( e_ord(1:ceil(0.95*length(e_ord))) ));% remove outliers
   thresh = outlier*sig_e;
   ibad = find(abs(e)>thresh)+P;
   igood = setdiff([1:M+P],ibad);

   % interpolate
   sh = x;
   sh(ibad) = -A(:,ibad)\(A(:,igood)*x(igood).');

   figure(r+1);
   subplot(211)
     plot([-P:M-1],sh,'-',[-P:M-1],s,'--r',[i-P-1],x(i),'.g',[-P:M-1],x,':k');
     axe = axis; axis([-P,M-1,axe(3:4)]);
     title('click restoration');
     legend('restored','ideal','corrupted',0);
   subplot(212)
     plot([0:M-1],abs(e),'g',[ibad-P-1],abs(e(ibad-P)),'.m');
     hold on; plot([-P,M-1],thresh*[1,1],'k:'); hold off;
     axe = axis; axis([-P,M-1,axe(3:4)]);
     title('click detection');
     legend('prediction error','detected','threshold',0);
   orient tall;
 end;
```