

Impulsantwort eines Systems bestimmen mit 'Maximum length sequences' und der Hadamard Transformation

Detlef Amberg 03/20
Detlef.Amberg@gmx.de

Fazit: Mit MLS ('maximum length sequence') lässt sich die Impulsantwort eines Systems ohne großen Rechenaufwand bestimmen.

Insbesondere ist keine FFT notwendig. Stattdessen wird für die Herleitung die Hadamard Transformation benutzt und im Ergebnis sind für die Berechnung der Impulsantwort aus der Systemantwort auf die MLS nur Addition/Subtraktion notwendig.

Der Algorithmus entstammt:

[Bor 1] 'An efficient Algorithmus for Measuring the Impulse Response Using Pseudorandom Noise', Jeffrey Borish, James B. Angell, J. Audio Eng. Soc., Vol. 31, No. 7, 1983 July/August

In

<http://www.mikrocontroller.net/topic/166117#new>

hatte ich die Vorgehensweise bereits dargestellt. Dort lag der Focus auf der Berechnung einer Korrelation mit dem Verfahren.

Es gilt aber (Formel 3 aus [Bor 1])

$$Dny = Dnn * h ,$$

Dny: Kreuzkorrelation zwischen Eingang und Ausgang

Dnn: Autokorrelation des Eingangssignals

h: Impulsantwort des Systems

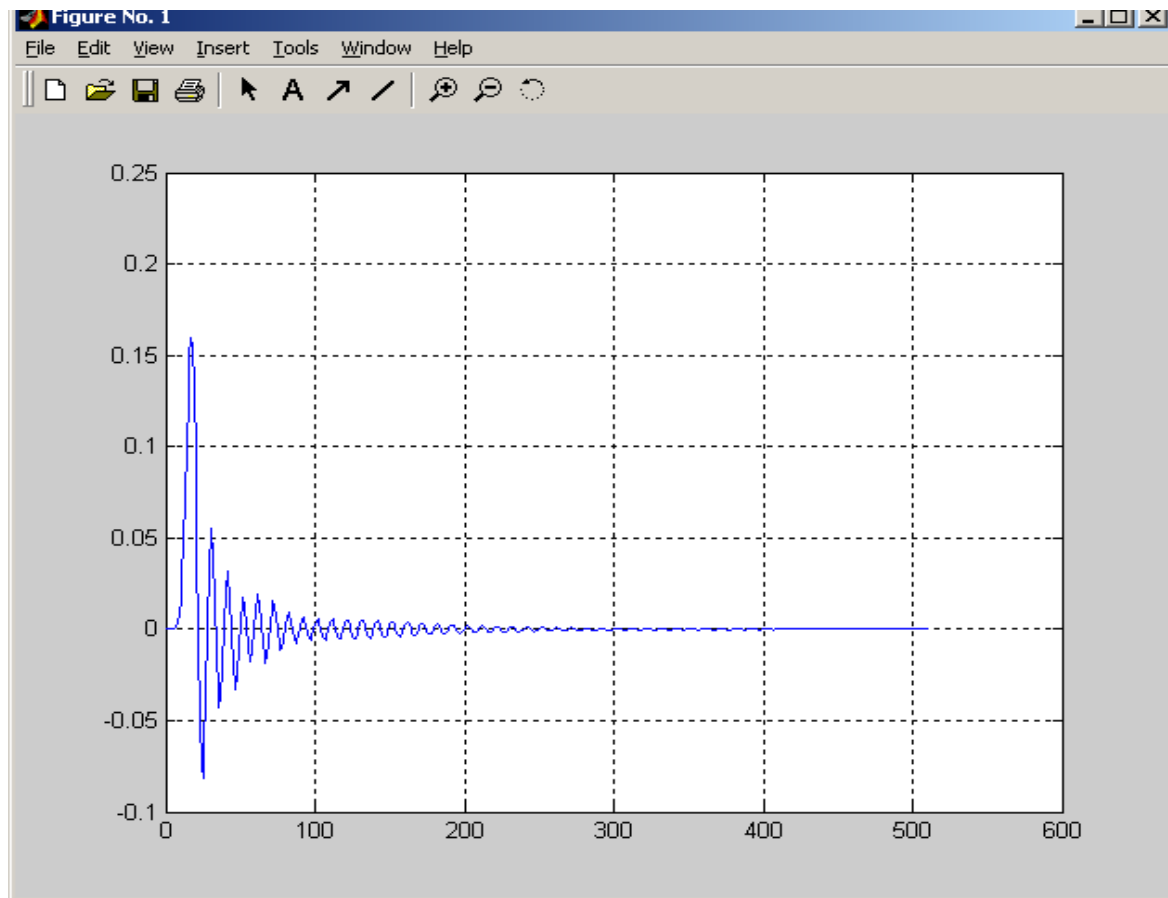
Für MLS ist die Dnn einem Dirac Impuls ähnlich. Die Dnn einer Folge der Länge $2^n - 1$ hat ein Maximum bei der Verschiebung 0 von $2^n - 1$, alle anderen Werte sind -1.

Damit ist die Kreuzkorrelation der MLS mit dem gemessenen Ausgangssignalk des Systems direkt die Impulsantwort des Systems. Die Kreuzkorrelation wird mit der Hadamard Transformation berechnet und reduziert sich auf einfache Operationen.

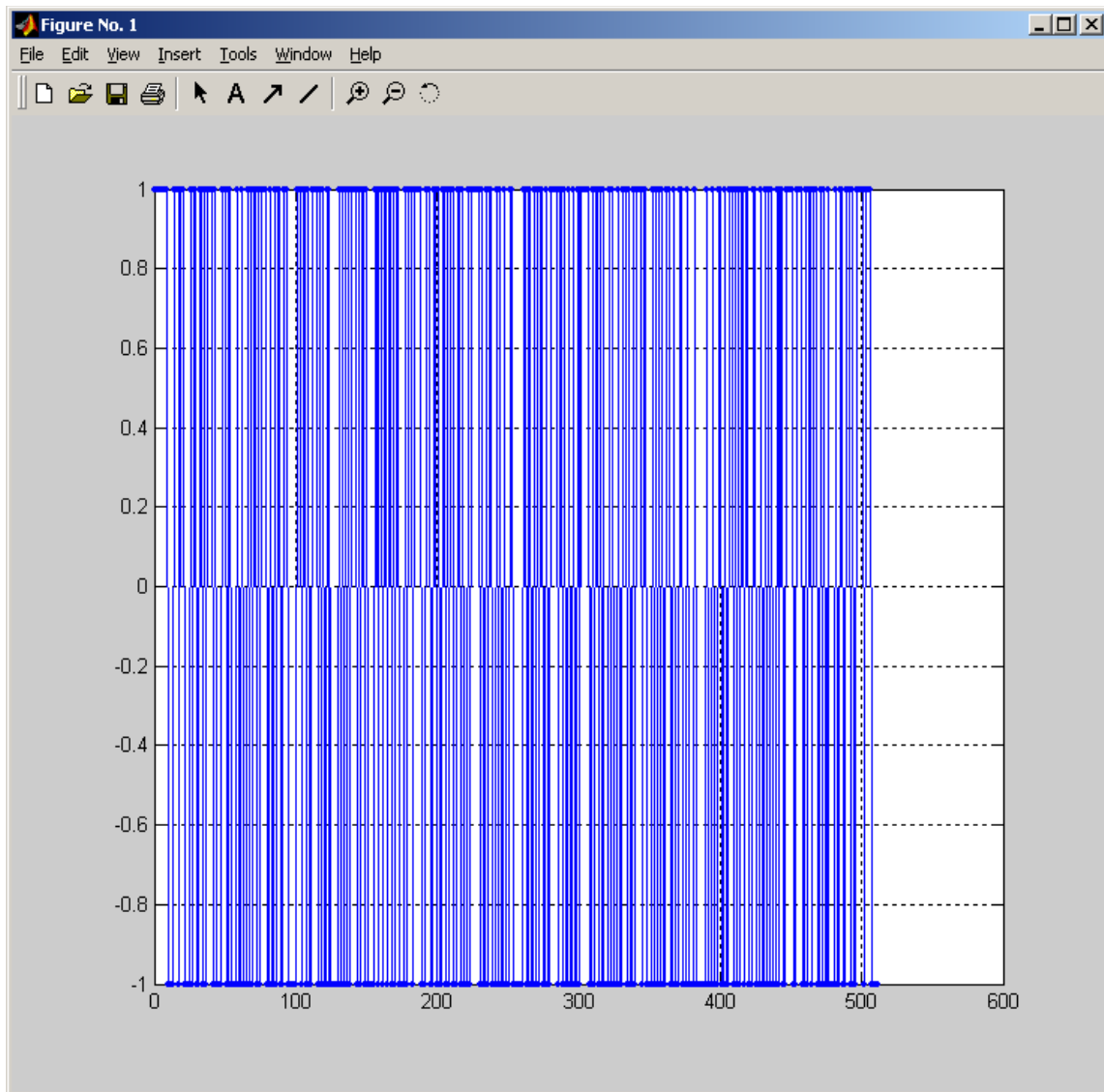
Ich habe das Verfahren in Matlab implementiert. Anhang A dokumentiert den Entwicklungscode, der verständlich ist aber nicht auf Platz optimiert wurde. Anhang B dokumentiert den Produktionscode, funktional weitgehend äquivalent und für Signale größerer Länge geeignet

Die Matlab Implementation

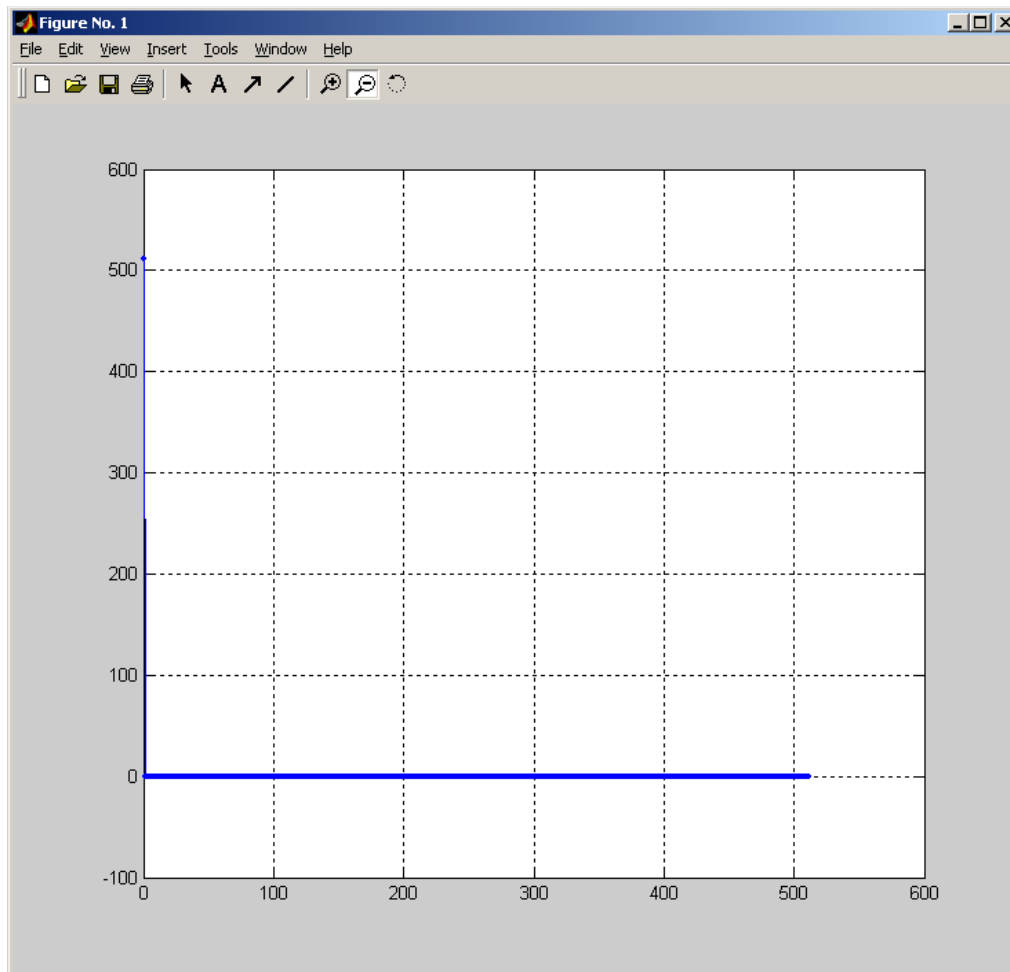
Der Matlab Code benutzt als Modellsystem ein Chebyshev Filter 10er Ordnung mit dieser Impulsantwort:



Die MLS Folge n der Länge 511 sieht so aus:

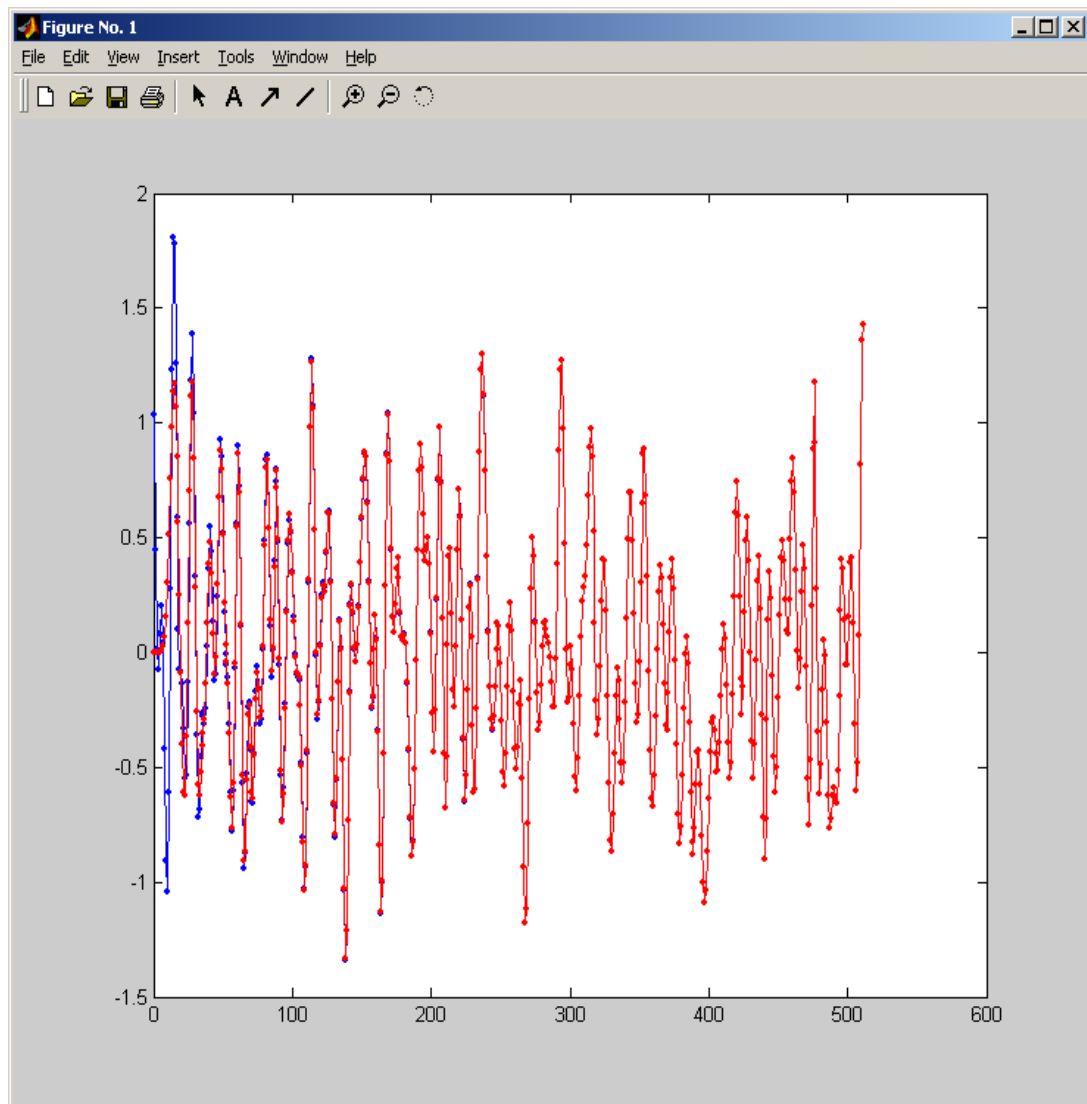


Die Autokorrelation Dnn dieser Folge:



Der erste Wert beträgt 511, alle anderen sind -1.

Die Antwort y_1 (rot) des Systems auf dieses Signal, die Faltung des Eingangssignals n mit der Impulsantwort in blau. Die beiden Signale unterscheiden sich durch das Einschwingen des Filters.



Aus y lässt sich die Impulsantwort h_r rückrechnen:

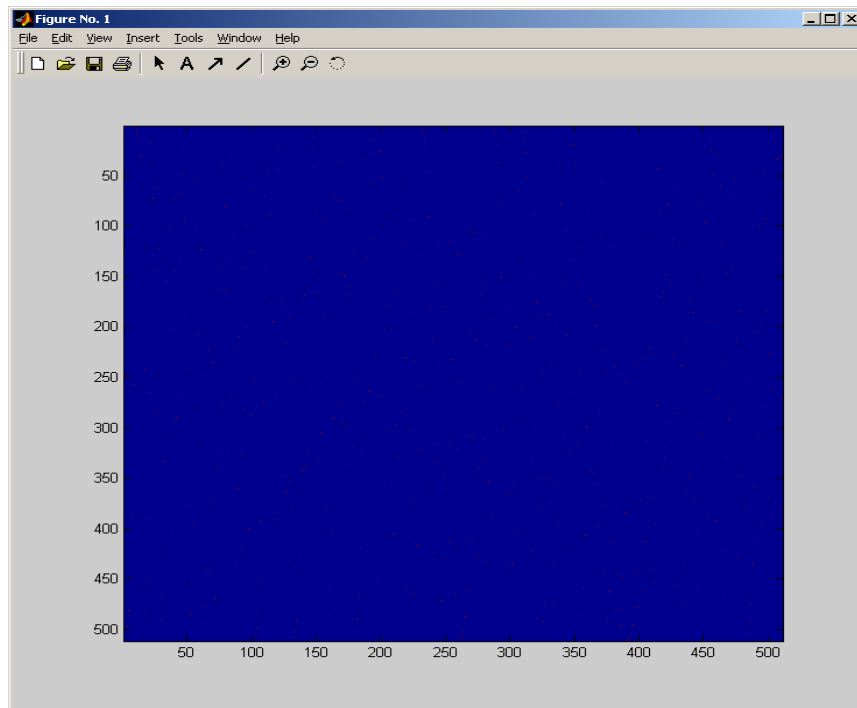
$$h_r = \text{inv}(MD_{nn}) * P_2 * S_2 * H * S_1 * P_1 * y;$$

$\text{inv}(MD_{nn})$: MD_{nn} (511*511) ist die Matrix der rotierten Autokorrelationen der MLS, die Hauptdiagonale ist mit 511 besetzt, alle anderen Werte sind -1. Die Invertierte ist ebenfalls eine symmetrische Diagonalmatrix. Diese Matrix entfaltet mit der Autokorrelation der MLS Folge, die kein perfekter Dirac ist. Die Auswirkungen auf nicht DC-freie Signale werden in obengenanntem paper untersucht. Für lange MLS wird diese Matrix gross und wird deswegen in der Produktionsversion nur insofern angewandt als sie das Ergebnis um einen Abtastwert verschiebt.

Algorithmus schlägt ohne die Invertierung der MLS für $\text{ord} > 3$ fehl, Grund unbekannt. Das ist der Grund für die Zeile im Algorithmus:

$$n = -n;$$

P_1, P_2 : Permutationsmatrizen, pro Zeile und Spalte ist ein Element 1, alle anderen sind 0

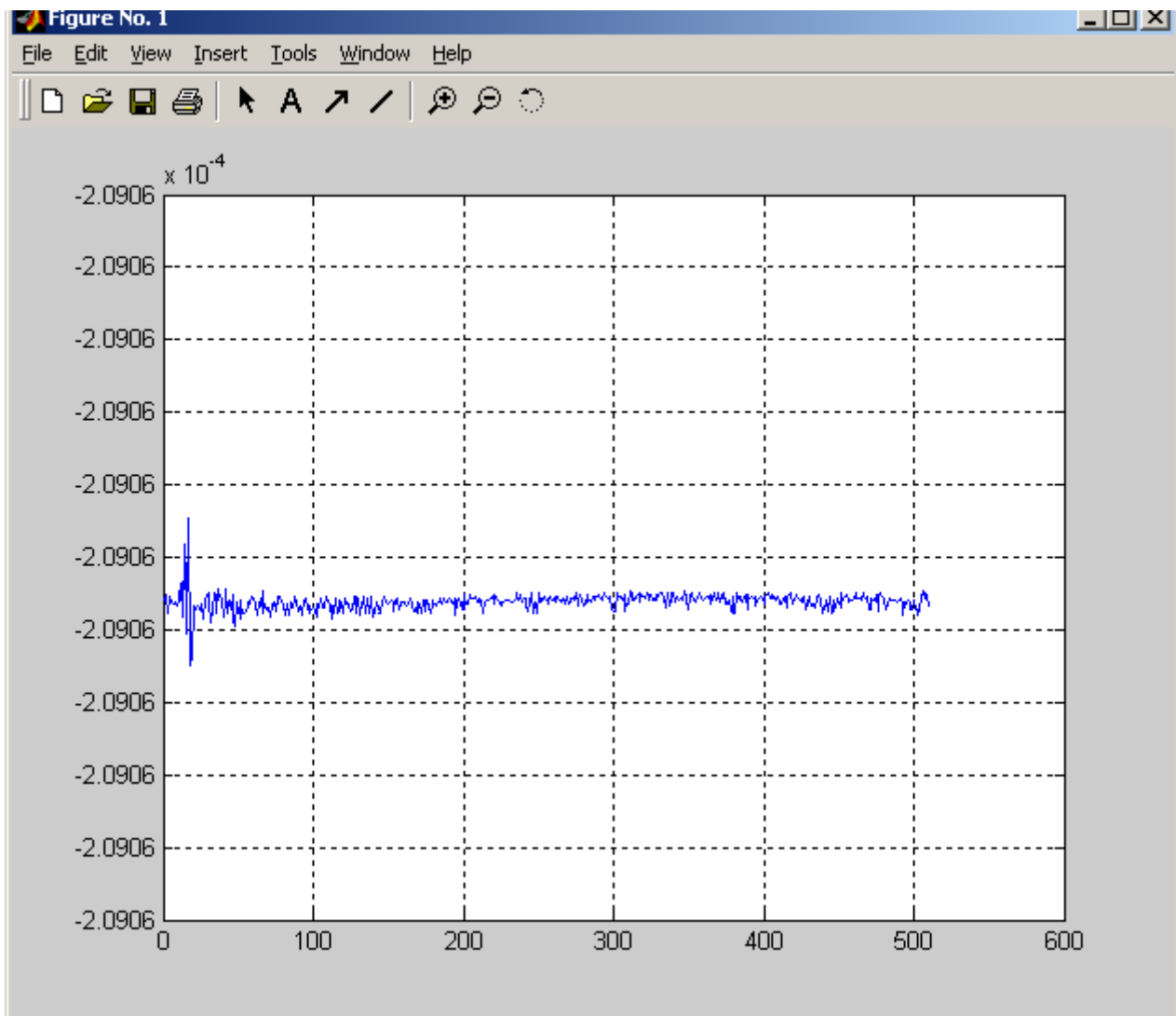


S_1 : Einheitsmatrix 511x511 mit 'darübersetzter' Zeile mit Nullen, wird zum 'Ausblenden' eines Elements benutzt. In der produktiven Implementation Streichen des ersten Vektorelements

S_2 : Einheitsmatrix 511x511 mit 'dahintergesetzter' Spalte mit Nullen, ähnlich S_1

H : Hadamard Matrix 512x512

Der Code entsprechend Anhang A rechnet die Impulsantwort richtig zurück (Fehler im Bereich der float-Rechengenauigkeit). Der Code entsprechend Anhang B verzichtet auf die Korrektur der nicht perfekten Autokorrelation der MLS. Das folgende Bild zeigt den Fehler der Rückrechnung:



Dieser Fehler ist vermutlich weiter reduzierbar. Mit wachsender Länge der MLS fällt er zunehmend weniger ins Gewicht. Eine vollständige Kompensation ist für lange MLS nicht möglich, da die entsprechende Matrix zu groß und der Rechenaufwand zu hoch wird.

Test des Algorithmus mit Meßsignalen, die um Bruchteile von Abtastwerten verschoben sind.

Es werden zwei Dinge getestet:

- Lässt sich die Impulsantwort rekonstruieren, wenn das Spektrum des Messsignals 'leckt', also das Signalfenster nichtganzzahlige Vielfache der Grundfrequenz enthält.
- Lässt sich die (verschobene) Impulsantwort rekonstruieren, wenn das Messsignal um den Bruchteil eines Abtastwertes verschoben ist.

Es stellt sich heraus, dass die Rekonstruktion in beiden Fällen möglich ist.

Das Testprogramm als Matlab script ist in Anhang C dokumentiert. Die Impulsantwort, die identifiziert werden soll, wird als bandbegrenztes Signal erzeugt. Das 'leaking' der Impulsantwort wird durch deren Kürzung erreicht, die Verschiebung um den Bruchteil eines Abtastwertes durch Multiplikation mit linearer Phase im Frequenzbereich. Das Messsignal wird durch Faltung mit der originalen beziehungsweise der verschobenen Impulsantwort erzeugt.

So sieht die Generierung der Impulsantworten aus:

```
h1 = exp(j*2*pi*randn(1,30));
h2 = h1.*exp(j*(1:30)*0.123456789e-1);

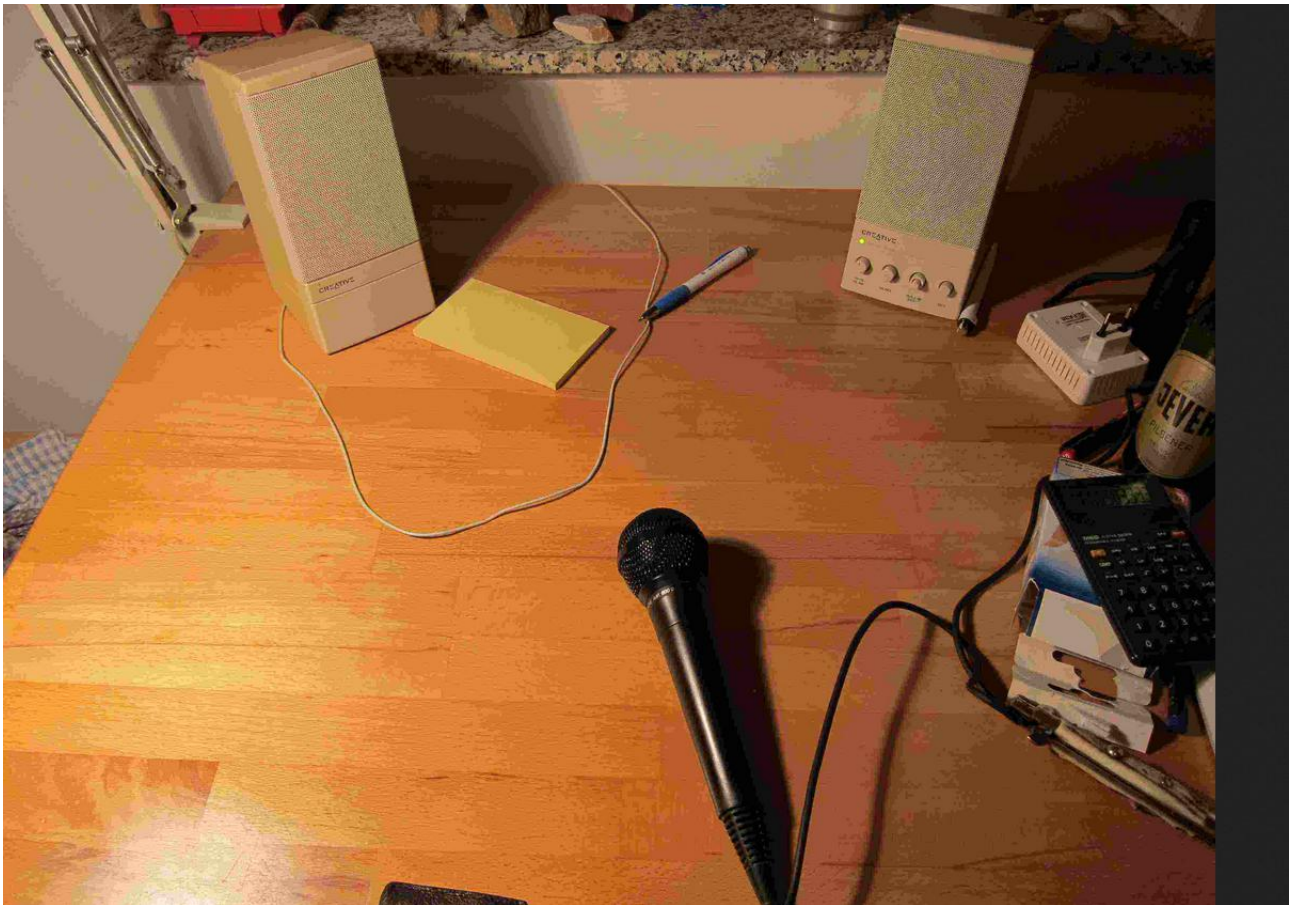
% der leckt bißchen raus
h1 = [zeros(1,30) h1 zeros(1,290-60)];
h2 = [zeros(1,30) h2 zeros(1,290-60)];

h1 = real(ifft([h1 conj(fliplr(h1(2:end))))));
h2 = real(ifft([h2 conj(fliplr(h2(2:end))))));

h1=h1(1:511);
h2=h2(1:511);
```

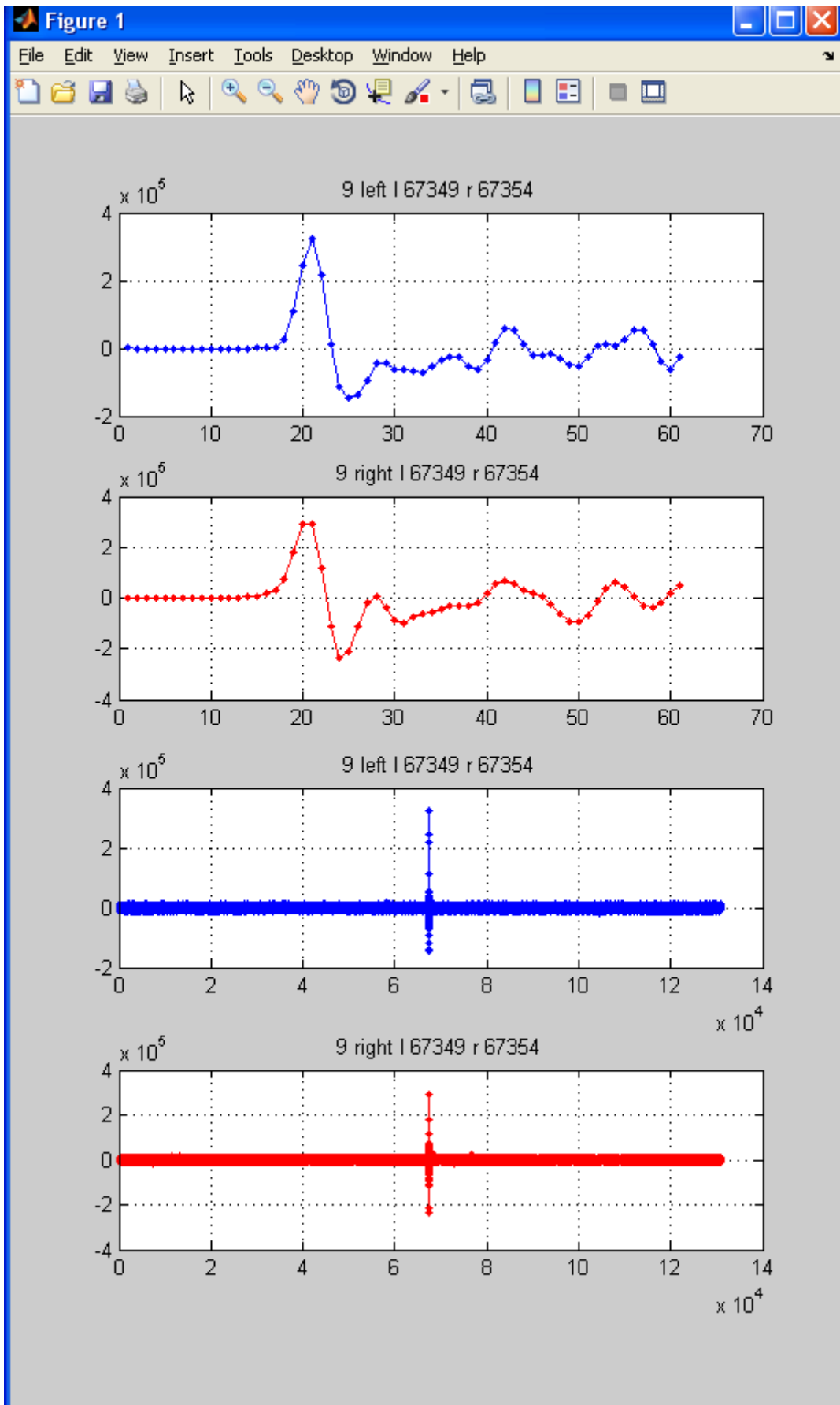
Ortung eines Mikrophons

Ich orte die Position eines Mikrophons mit dem Verfahren.



Die beiden Lautsprecher werden mit zwei unterschiedlichen $2^{17}-1$ Abtastwerten langen MLS beaufschlagt. Die pre/postsort Permutationen werden vorab berechnet. Anhang D zeigt den Matlab-Code. Bei einer Abtastfrequenz von 48kHz und einer angenommenen Schallgeschwindigkeit von 331m/s ergibt sich eine Auflösung von ca. 0.7cm. Mit einer höheren Abtastrate und einer Interpolation der Impulsantwort lässt sich die Auflösung steigern. Das Verfahren funktioniert auch, wenn man die MLS mit einem Hochpass vorfiltert, sodass das Audiosignal unhörbar wird.

Das folgende Bild zeigt die linke und die rechte Impulsantwort in blau und rot, oben herausgezoomt.



Die Position und Form der berechneten Impulsantworten ist absolut stabil.

Anhang A

```
clear

ord=9;
zho=2^ord;
zhom1=zho-1;

n=prnum(ord);
n=2*n-1;
n=n(1:end-1);
% Vorsicht, wenns nicht klappt am
% minus rumfummeln
n=-n;

Dnn=corr(n,n);

h=randn(1,length(n));
%h=1:length(n);
h=h(:);

y=myconv(n,h);
Dny=corr(n,y);
N=zeros(zhom1,zhom1);
ii=n(:).';
%nein ii=fliplr(ii);
for(k=1:length(ii))
    N(k,:)=ii;
    ii=[ii(end) ii(1:end-1)]; %linksrum/rechtsrum?
end

myconv(Dnn,h);
% corr(n,y) so nicht !
%corr(y,n)

S1=[zeros(1,zho-1) ;eye(zho-1)];
S2=S1.';

H=hadamard(zho);
G=H(2:end,2:end);
Ns=((1-N)/2);

p_N=zeros(ord,zhom1);
ii=n(:).';
for(k=1:ord)
    p_N(k,:)=ii;
    ii=[ii(end) ii(1:end-1)];
end
p_N=((1-p_N)/2);

%iaa=2.^(0:ord-1)*Ns(1:ord,:);
iaa=2.^(0:ord-1)*p_N(1:ord,:);
[iic,iib]=sort(iaa);
```

```

presort=iib;
Nst=Ns(:,iib);
P1=inv(inv(Ns)*Nst);

%P2=N*inv(P1)*inv(G);
%P2=N*inv(P1)*(G'-1)/zho;

P2s=N*P1'*(G'-1)/zho;

p_N=zeros(zhom1,ord);
ii=n(:).';
for(k=1:zhom1)
    bla=ii;
    bla=bla(presort);
    bla=(1-bla)/2;
    p_N(k,:)=bla(2.^(0:ord-1));
    ii=[ii(end) ii(1:end-1)];
end
bla=p_N*(2.^(0:ord-1).');

iii=Nst(:,2.^(0:ord-1))*(2.^(0:ord-1).');
[iik,iim]=sort(iii);
Gs=Nst(iim,:);
P2=inv(Gs*inv(Nst));
%postsort=iii;
postsort=bla;

%P2=P2s;

nn=fliplr(Dnn. ');
MDnn=nn;
for(k=1:length(nn)-1)
    nn=[nn(end) nn(1:end-1) ];
    MDnn= [MDnn;nn];
end;

%iid=P2*S2*H*S1*P1*y;
iif=[0 ;y(presort)];
iig=hadamard_butterfly(iif);
iih=iig(2:end);
%iid=P2*iih;
iid=iih(postsort);

iie=(iid-mean(iid))/zho;
iie=[iie(2:end);iie(1)]; %1 mal rotieren
hr=inv(MDnn)*iid;

sum(abs(hr-h))
sum(abs(iie-h))
%sum(abs(iid/zho-h))
plot([h hr iie])
sum(sum(abs(P2'*N*P1'-G)))
sum(sum(abs(P2'*Ns*P1'-Gs)))

```

```
sum(sum(abs(Ns-((1-N)/2))))  
sum(sum(abs(Gs-((1-G)/2))))  
return
```

Anhang B

```
clear

ord=17;
zho=2^ord;
zhom1=zho-1;

n=prnum(171);
n=2*n-1;
n=n(1:end-1);
% Vorsicht, wenns nicht klappt am
% minus rumfummeln
n=-n;

h=randn(1,length(n));
%h=1:length(n);
h=h(:);

y=myconv(n,h);

p_N=zeros(ord,zhom1);
ii=n(:).';
for(k=1:ord)
    p_N(k,:)=ii;
    ii=[ii(end) ii(1:end-1)];
end
p_N=((1-p_N)/2);

%iaa=2.^(0:ord-1)*Ns(1:ord,:);
iaa=2.^(0:ord-1)*p_N(1:ord,:);
[iic,iib]=sort(iaa);
presort=iib;

p_N=zeros(zhom1,ord);
ii=n(:).';
for(k=1:zhom1)
    if(mod(k,100)==0)
        zho-k
    end;
    bla=ii;
    bla=bla(presort);
    bla=(1-bla)/2;
    p_N(k,:)=bla(2.^(0:ord-1));
    ii=[ii(end) ii(1:end-1)];
end
bla=p_N*(2.^(0:ord-1).');

postsort=bla;

%iid=P2*S2*H*S1*P1*y;
iif=[0 ;y(presort)];
iig=hadamard_butterfly(iif);
```



```
iih=iig(2:end);  
%iid=P2*iih;  
iid=iih(postsort);  
  
iie=(iid-mean(iid))/zho;  
iie=[iie(2:end);iie(1)]; %1 mal rotieren  
  
%hr=inv(MDnn)*iid;  
  
sum(abs(iie-h))  
%sum(abs(iid/zho-h))  
plot([h iie])
```

Anhang C

```
clear

ord=9; % Laenge der MLS (2^ord)-1
zho=2^ord;

h1 = exp(j*2*pi*randn(1,30));
h2 = h1.*exp(j*(1:30)*0.123456789e-1);

%h1 = [zeros(1,30) h1 zeros(1,256-60)];
%h2 = [zeros(1,30) h2 zeros(1,256-60)];

% der leckt bißchen raus
h1 = [zeros(1,30) h1 zeros(1,290-60)];
h2 = [zeros(1,30) h2 zeros(1,290-60)];

h1 = real(ifft([h1 conj(fliplr(h1(2:end))))));
h2 = real(ifft([h2 conj(fliplr(h2(2:end))))));

h1=h1(1:511);
h2=h2(1:511);

plot([h1 ; h2].')

sph1=fft(h1);
sph2=fft(h2);
nnn=40;
disp(sprintf('Phasenversch. Impulsantwort %.10e',....
    angle(sph1(nnn)./sph2(nnn))));

n=prnum(ord);
n=n(1:end-1);
n=(2*n)-1; % DC rausmachen, fast

h=h2;

Dnn=corr(n,n);
y=myconv(h,n);

Dny=corr(y,n);

nn=fliplr(Dnn. ');
MDnn=nn;
for(k=1:length(nn)-1)
    nn=[nn(end) nn(1:end-1) ];
    MDnn= [MDnn;nn];
end;

nn=n;
N=nn;
for(k=1:length(nn)-1)
    nn=[nn(end) nn(1:end-1) ];
```

```

        N= [N;nn];
end;

%N*y
H=hadamard(2^ord);
G=H(2:end,2:end);

Ns=(1-N)/2;
sP1=(Ns(1:ord,:).')*(2.^(0:(ord-1))).';
[ff,sP1]=sort(sP1);
P1=zeros(zho-1,zho-1);
P1((0:zho-2)*(zho-1)).'+sP1)=1;
P1=P1.';

bla=Ns*P1.';
bla=N *P1.';

P2=(G*inv(bla)).';
S1=[zeros(1,zho-1) ;eye(zho-1)];
S2=S1.';

hr=inv(MDnn)*P2*S2*H*S1*P1*y;
hr=hr(:);
sphr=fft(hr);
disp(sprintf('Phasenversch. Impulsantwort %.10e',....
    angle(sphr(nnn)./sph1(nnn))));
disp(sprintf('Phasenversch. Impulsantwort %.10e',....
    angle(sphr(nnn)./sph2(nnn))));

plot(1:length(h),h,'b.-',1:length(hr),hr,'r.-');

return

```

Anhang D

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear

load c:\proj\hadamard\impulse_prn17.mat
mls_left = n;
postsort_left = postsort;
presort_left=presort;

load c:\proj\hadamard\impulse_prn17a.mat
mls_right = n;
postsort_right = postsort;
presort_right=presort;

mml=mls_left;
mmr=mls_right;

zhoml=2^17-1;

fs=48000;
%sec=5;
n=fix(2.5*zhoml/fs);
n=2;
mml=repmat(mml,1,n);
mmr=repmat(mmr,1,n);

[fb,fa]=butter(5,0.5,'high');
%mml=filter(fb,fa,mml);
%mmr=filter(fb,fa,mmr);
tx=[ mml ;mmr].';

kk=1;

while(1)

soundsc(tx,fs);
rec=wavrecord(length(tx), fs, 'int16');

%pause(length(tx)/fs+1);

disp('Z')

rec=double(rec);
rec=rec-mean(rec);

von=fix(length(rec)/4);
rec=rec(von:end);

rec=rec(1:zhoml);

disp('A')
yr=double(rec);
%yr=double(rec(end-1000:end-1000+510));

y=yr;
bla= hadamard_butterfly([0 ; y(presort_right)]);
%bla= fwht([0 ; y(presort_right)]);
hr=bla(2:end);
hr=hr(postsort_right);

disp('B')
```

```

y=(yr);
bla= hadamard_butterfly([0 ; y(presort_left)]);
%bla= fwht([0 ; y(presort_left)]);
disp('C')
hl=bla(2:end);
hl=hl(postsort_left);
disp('D')

[bla,indr]=max(hr);
[bla,indl]=max(hl);

subplot(4,1,1)
hlp=hl(indl-20:indl+40);
plot(1:length(hlp),hlp,'b.-');
title(sprintf('%d left 1 %d r %d',kk,indl,indr))
%axis([0 511 250 480])
grid
subplot(4,1,2)
hrp=hr(indr-20:indr+40);
plot(1:length(hrp),hrp,'r.-');
title(sprintf('%d right 1 %d r %d',kk,indl,indr))
%axis([0 511 250 480])
grid
subplot(4,1,3)
hlp=hl;
plot(1:length(hlp),hlp,'b.-');
title(sprintf('%d left 1 %d r %d',kk,indl,indr))
%axis([0 511 250 480])
grid
subplot(4,1,4)
hrp=hr;
plot(1:length(hrp),hrp,'r.-');
title(sprintf('%d right 1 %d r %d',kk,indl,indr))
%axis([0 511 250 480])
grid

pause
kk=kk+1;
end;

return

```