ELEC2021 laboratory exercises
Signal processing with Matlab

Ivan Markovsky

Preparation

- **Matlab:** Read the tutorial, available at

  `http://www.maths.dundee.ac.uk/~ftp/na-reports/MatlabNotes.pdf`

  Start Matlab and try out simple examples (*e.g.*, the ones in the tutorial). Use "`help fun`" to get online help of the usage of the function `fun`. Use an editor, *e.g.*, Matlab's editor (type `edit`), to write script files for examples, requiring more than a few commands.

- **Background reading:** Revise the material on difference/differential equations, convolution, and Fourier series.

- **Practise problems:** Solve the following problems:

  – Find the solution to the difference equation

  $$y(t) - \frac{1}{4}y(t-2) = 0,$$

  under initial conditions $y(-1) = 1$ and $y(-2) = 0$.

  Solution: Observe that the difference equation is homogeneous, linear, with constant coefficients. In this (simplest possible case) case, the solution is is of the form $y(t) = \sum_i c_i z_i^t$, where z_i are the solutions of the characteristic equation and c_i are constants that depend on the initial conditions. The characteristic equation is

  $$z^2 - \frac{1}{4} = \left(z - \frac{1}{2}\right)\left(z + \frac{1}{2}\right) = 0 \tag{1}$$

  and its solutions are

  $$z_1 = -\frac{1}{2} \qquad \text{and} \qquad z_2 = \frac{1}{2}.$$

  Therefore, the general solution of (1) is of the form

  $$y(t) = c_1\left(-\frac{1}{2}\right)^t + c_2\left(\frac{1}{2}\right)^t, \qquad \text{for } t \geq 0. \tag{2}$$

  It order to find the constants c_1 and c_2, we need to fit the general solution to the given initial conditions. Since the general solution is valid for $t \geq 0$, we compute

  $$y(0) = \frac{1}{4}y(-2) = 0 \qquad \text{and} \qquad y(1) = \frac{1}{4}y(-1) = \frac{1}{4}.$$

  Then, from (2) we have the following system of equations for c_1 and c_2

  $$y(0) = 0 = c_1 + c_2$$
  $$y(1) = \frac{1}{4} = \frac{1}{2}(-c_1 + c_2),$$

which solution is $c_1 = -1/4$ and $c_2 = 1/4$. Therefore,

$$y(t) = \frac{1}{4}\left(\left(\frac{1}{2}\right)^t - \left(-\frac{1}{2}\right)^t\right).$$

$\square$

- Let s be the unit step

$$s(t) = \begin{cases} 0, & \text{for } t \leq 0 \\ 1, & \text{for } t > 0. \end{cases} \tag{3}$$

  Find the discrete-time convolution

$$y = h \star u = \sum_{\tau=0}^{\infty} h(t)u(t-\tau),$$

  where $h(t) = a^t s(t)$, for some $a \in \mathbb{R}$, and $u = s$.

  Solution:

$$\begin{aligned} y(t) = (h \star u)(t) &= \sum_{\tau=-\infty}^{\infty} h(\tau)u(t-\tau) \\ &= \sum_{\tau=-\infty}^{\infty} a^\tau s(\tau)s(t-\tau) \\ &= \begin{cases} \sum_{\tau=0}^{t} a^\tau = \frac{1-a^{t+1}}{1-a}, & \text{for } t \geq 0 \\ 0, & \text{for } t < 0 \end{cases} \\ &= \frac{1-a^{t+1}}{1-a} s(t) \end{aligned}$$

$\square$

- Find the Fourier transform of the rectangular pulse signal

$$p(t) = \begin{cases} 1, & \text{for } |t| \leq 0.5 \\ 0, & \text{for } |t| > 0.5. \end{cases} \tag{4}$$

  Solution:

$$\begin{aligned} P(\omega) &= \int_{-\infty}^{\infty} p(t)e^{-\mathbf{i}\omega t}\mathrm{d}t = \int_{-0.5}^{0.5} e^{-\mathbf{i}\omega t}\mathrm{d}t \\ &= -\frac{1}{\omega\mathbf{i}}\left(e^{-\mathbf{i}\omega 0.5} - e^{-\mathbf{i}\omega 0.5}\right) \\ &= -\frac{2}{\omega}\frac{1}{2\mathbf{i}}\left(e^{-\mathbf{i}\omega 0.5} - e^{-\mathbf{i}\omega 0.5}\right) = \frac{2}{\omega}\sin(\omega 0.5) \end{aligned}$$

$\square$

Exercises

1. *Generation and plotting of discrete-time signals*   Explain what does the following code:

```
clear all, close all
T = 1; ts = 0.01; % time horizon and discretization time
t = 0:ts:T;       % time vector
u = sin(2*pi*t);  % signal

figure, hold on, stem(t,u,'k')
title('Signal'), xlabel('time, t'), ylabel('u(t)')
```

   Execute it and observe the effect of modifying the `T` and `ts` parameters.

   Solution: The code generates two vectors—one of equidistantly spaced points in the interval $[0,T]$, and one of the values of the function $u(t) = \sin(2\pi t)$ at the points t that are elements of the first vector—and plots the first vector versus the second. As a result we get an approximation of the graph of the function u over the interval $[0,T]$. Obviously, the approximation is better when the points in the first vector get closer to each other. This corresponds to choosing the discretization step smaller.

   The plot generated by the code is shown in Figure 1.

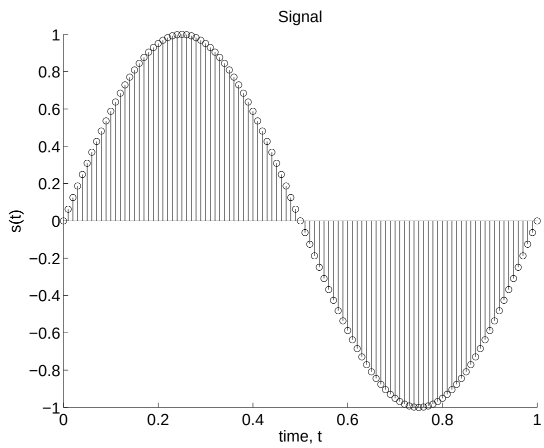

Figure 1: The figure generated by the code in Exercise 1.

□

2. *Time and phase shift*   Assuming that the code of Exercise 1 is executed first, explain what does the following code:

```
u1 = sin(2*pi*(t+1/4)); % positive time shift
u2 = sin(2*pi*(t-1/4)); % negative time shift
u3 = sin(2*pi*t+pi/2);  % positive phase shift
u4 = sin(2*pi*t-pi/2);  % negative phase shift

figure, hold on
stem(t,u,'k'), stem(t,u1,'r'), stem(t,u2,'b')
title('Original signal and its time shifts')
xlabel('time, t'), ylabel('u(t), u1(t), u2(t)')

figure, hold on
stem(t,u,'k'), stem(t,u3,'r'), stem(t,u4,'b')
```

```
title('Original signal and its phase shifts')
xlabel('time, t'), ylabel('u(t), u3(t), u4(t)')
```

Comment on the results.

Solution: A time shift by τ is equivalent to a phase shift by $\phi := \omega\tau$ radians, where ω is the frequency. Indeed,

$$\sin\big(\omega(t+\tau)\big) = \sin(\omega t + \underbrace{\omega\tau}_{\phi}) = \sin(\omega t + \phi).$$

Vice verse, a phase shift by ϕ radians is equivalent to a time shift by $\tau := \phi/\omega$. Positive time/phase shift has the effect of moving the original function's graph to the left and negative time/phase shift to the right. The two plots generated by the code are identical and are shown in Figure 2.

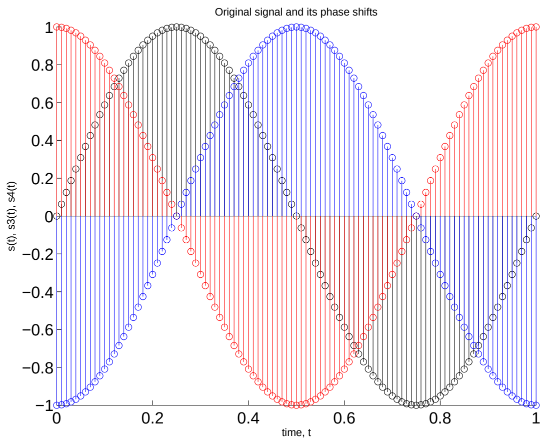

Figure 2: The figures generated by the code in Exercise 2.

□

3. *Time scaling* Assuming that the code of Exercise 1 is executed first, explain what does the following code:

```
u5 = sin(2*pi*(2*t));   % time scale by constant > 1
u6 = sin(2*pi*(1/2*t)); % time scale by constant < 1

figure, hold on
stem(t,u,'k'), stem(t,u5,'r'), stem(t,u6,'b')
title('Original signal and its time scaled versions')
xlabel('time, t'), ylabel('u(t), u5(t), u6(t)')
```

Comment on the results.

Solution: Time scaling changes the frequency. Scaling by a factor bigger than one increases the frequency, which has the effect of shrinking the original signal. Scaling by a factor smaller than one decreases the frequency, which has the effect of expanding the original signal.

□

4. *Homogeneous difference equations* Write a function that computes the first $T = 10$ samples of the solution of the homogeneous difference equation

$$y(t) = y(t-1) + y(t-2), \qquad \text{for} \quad t = 1, 2, \dots, T,$$

with initial conditions

$$y(-1) = 0, \quad y(0) = 1.$$

(Note: The elements of the resulting sequence are called the Fibonacci numbers, see

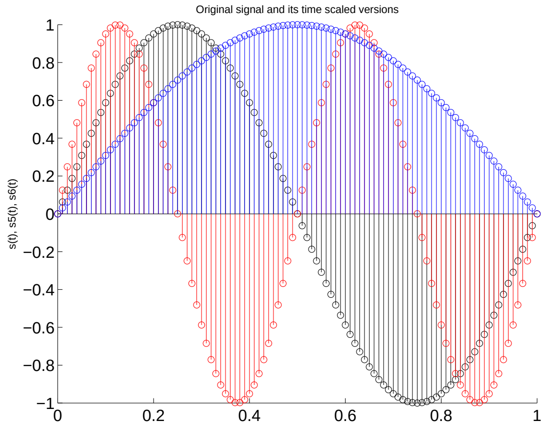

Figure 3: The figures generated by the code in Exercise 3.

`http://en.wikipedia.org/wiki/Fibonacci_number`)

Solution: Two Matlab functions—iterative and recursive—are:

```
% Iterative version

function y = fibonacci1(T)
y(1) = 0; y(2) = 1; % initial conditions
for t = 1:T
  y(t+2) = y(t+1) + y(t);
end

% Recursive version

function y = fibonacci2(T)
if T == 0
  y = [0 1];
else
  y = fibonacci2(T-1);
  y = [y y(end-1)+y(end)];
end
```

Result:

```
>> y = fibonacci1(10)
y = 0    1    1    2    3    5    8   13   21   34   55   89
>> y = fibonacci2(10)
y = 0    1    1    2    3    5    8   13   21   34   55   89
```

$\square$

5. *Non-homogeneous difference equations* Describe qualitatively the solution of the difference equation

$$y(t) - 0.5y(t-1) = \delta(t),$$

with zero initial condition, where δ is the unit pulse

$$\delta(t) = \begin{cases} 1, & \text{when } t = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Compute the solution numerically and compare with your prediction. (You can use Matlab's function `filter`.)

Solution: From a signal processing point of view, the solution of a nonhomogenious difference equation is a response of a filter to an input signal (given by the nonhomogenious part of the equation). Since the input is a pulse and the initial conditions are zero, the response of the filter is the impulse response. The filter has a single pole at $z = 0.5$, so that the impulse response is an exponential $h(t) = c(0.5)^t$ for $t = 1, 2, \dots$

The solution can be computed numerically and plotted by following commands:

```
T = 8; % simulation time
u = [1 zeros(1,T)];
h = filter(1,[1 -0.5],u);
figure, stem(0:T,h,'k')
xlabel('time, t'), ylabel('h(t)')
```

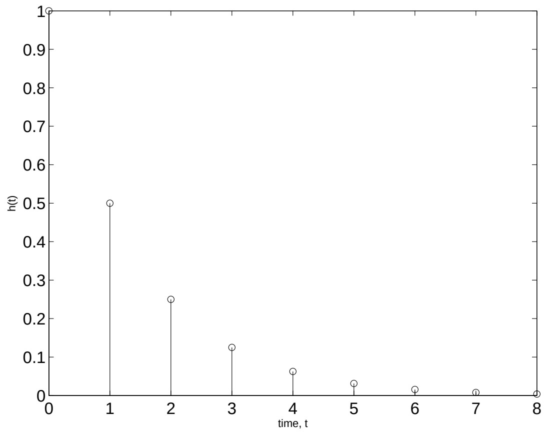

Figure 4: Impulse response of the the filter in Problem 5.

□

6. *Convolution* Compute the convolution of the impulse response, obtained in Problem 5, with the unit step signal s, see (3). (You can use Matlab's function `conv`.) What do you obtain? Confirm your answer.

   Solution: Assuming that the impulse response is already computed and is stored in a vector `h`, the following code computes the convolution of `h` with the unit step

```
u = [0 ones(1,T)];
y = conv(h,u);
```

   The result is the step response. Indeed,

```
s = filter(1,[1 -0.5],[0 ones(1,T)]);
norm(s - y(1:T+1))

ans = 0
```

□

7. *Response of a linear time-invariant system to a sine input* Describe qualitatively the solution of the differential equation

$$y(t) + \frac{\mathrm{d}}{\mathrm{d}t} y(t) = \sin(t). \tag{5}$$

Verify your answer numerically. (You can use Matlab's function `lsim`.)

Solution: From a signal processing point of view, the differential equation (5) describes a continuous-time linear time-invariant filter subjected to an input $y(t) = \sin(t)$. You know that the steady state response of a linear time-invariant system to a input $y(t) = \sin(\omega t)$ is

$$y(t) = |H(\mathbf{i}\omega)| \sin(\omega t + \angle H(\mathbf{i}\omega)),$$

where H is the transfer function of the system. Therefore, the answer to the question "describe qualitatively the solution" is

> "a sine function with the same frequency, *i.e.*, $c\sin(t+\phi)$, for some c and ϕ.".

Numerical verification:

```
T = 3*pi; t = 0:0.01:T; u = sin(t);
y = lsim(tf(1,[1 1]),u',t);

figure, plot(t,y,'k')
xlabel('time, t'), ylabel('y(t)')
```

The resulting plot is shown in Figure 5. Observe the transient in the beginning of the response.

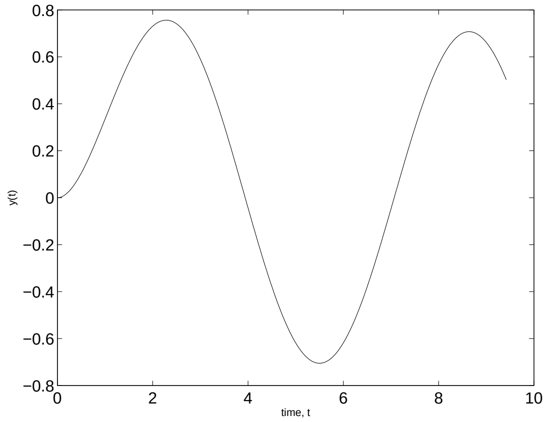

Figure 5: Response of the filter in Problem 7.

□

8. *Approximation of a signal by finite Fourier series; Gibbs phenomenon* Find approximations of the rectangular pulse signal p, see 3, by truncating its Fourier series to the first $1, 3, 5, 7, 9$, and 11 complex exponentials. (You can use the functions `fft` and `ifft`.)

   Solution:

```
t = -1:0.01:1; T = length(t);
p = zeros(1,T);
p(intersect(find(t >= -0.5),find(t <= 0.5))) = 1;
fp = fft(p);
figure, hold on, plot(t,p)
for i = 0:2:10
  ph = real(ifft([fp(1:i+1) zeros(1,T-2*i-1) fp(end-i+1:end)]));
  plot(t,ph,'--'), % pause
end
```

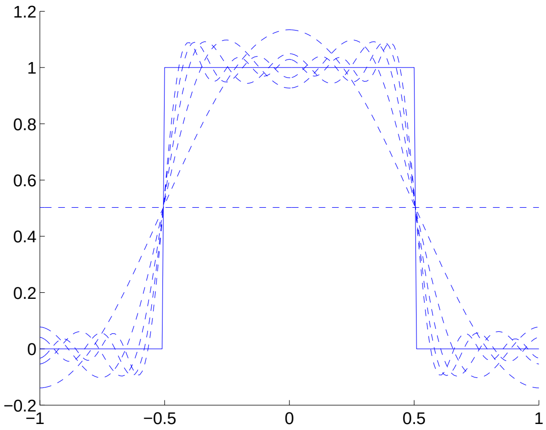

Figure 6: Approximation of the rectangular pulse by truncated Fourier series.

Observe that by increasing the number of exponentials, the peak error $\max_t |p(t) - \widehat{p}(t)|$ does not drop significantly. In fact, even asymptotically when the number of exponentials going to infinity, the error does not go below 8.9%. This fact is known as Gibbs phenomenon. □

9. *Delay estimation by cross-correlation* A signal u is sent out, takes time τ to reach an object, and is reflected back and received. In total, the signal is delayed by a time 2τ, attenuated by an amount a, and is subject to noise n. Thus the received signal is modeled by

$$y(t) = au(t - 2\tau) + n(t).$$

You know u and y, but because of the noise you are uncertain about τ, which is what you want to know, to determine the distance of the object from the sender/receiver. You also don't know the noise, but you can assume that u and n are uncorrelated. Explain how to find τ. Apply your method on the data available from

`http://users.ecs.soton.ac.uk/im/elec2021/delay.mat`

Solution:

```
% data
if 0
  T   = 1000;
  u   = randn(1,T);
  y0  = [0 0 u(1:end-2)];
  y   = y0 + 0.1 * randn(1,T);
  save delay.mat u y
else
  load('delay.mat')
end


% method
ryu = xcorr(y,u);
M   = round(length(ryu)/2); t = 10;
figure, plot(-t:t,ryu(M-t:M+t)), grid on
set(gca,'xtick',-10:10)
xlabel('t'), ylabel('ruy(t)')
```

□

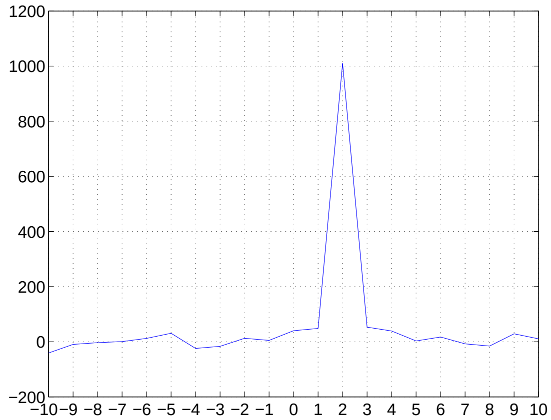

Figure 7: Input/output cross-correlation function.

10. *Filtering* Consider a received signal

$$y(t) = u(t) + n(t),$$

where u is the transmitted signal and n is a zero mean white Gaussian noise. The transmitted signal is smooth compared with the noise. Using this knowledge and the fact that the noise is zero mean white Gaussian, explain how to improve the signal-to-noise ratio of the received signal. Apply your method on the data available from

```
http://users.ecs.soton.ac.uk/im/elec2021/filtering.mat
```

Solution:

```
% data
if 0
  T = 1; ts = 0.0004;
  t = 0:ts:T;
  u = sin(4*pi*t);
  y = u + 2 * randn(size(u));
  %figure, hold on, plot(t,y,'-k')
  save filtering.mat t u y
else
  load('filtering.mat')
end

% method
M = 250; b = ones(1,M)/M; a = 1;
uh = filter(b,a,y);
plot(t,uh,'--b')

D = M/2;
ts  = t(1:end-D+1);
us  = u(1:end-D+1);
ys  = y(1:end-D+1);
uhs = uh(D:end);

figure, hold on
plot(ts,us,'-r')
plot(ts,uhs,'--b')
legend('u','uh') % 'y'
```

```
xlabel('time, t'), ylabel('u(t), y(t)')

uh_snr = norm(us) / norm(us - uhs)
y_snr  = norm(us) / norm(us - ys)
```

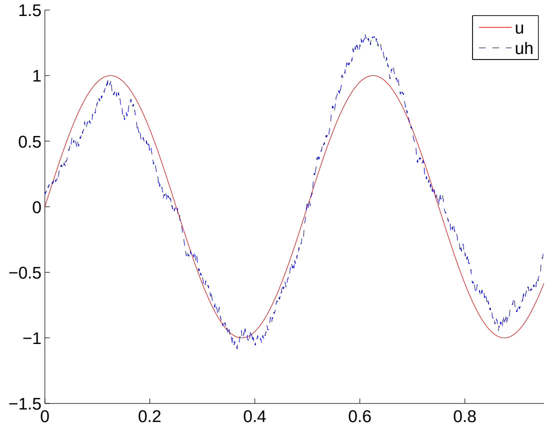

Figure 8: True and estimated signals.

□

Additional problems

1. *Sum of sines* Let

$$f(t) = \sin(2\pi mt) + \sin(2\pi nt),$$

where m and n are positive integers. Is f periodic? If so, what is its period?

Hint: Choose particular positive integers m, n, and plot the resulting function f to see if its periodic. Of course, you can not draw general conclusions from the examples (why?) but the examples may *suggest* what is the answer to the general question, so that you know what you are aiming to prove.

Solution: The function f is periodic. Let d be the greatest common divisor of m and n (*i.e.*, the greatest integer d, for which there are integers p and q, such that $m = pd$ and $n = qd$) and let $T = 1/d$. Then

$$\begin{aligned}
f(t+T) &= \sin(2\pi mt + 2\pi mT) + \sin(2\pi nt + 2\pi nT) \\
&= \sin(2\pi mt + 2\pi m\frac{p}{m}) + \sin(2\pi nt + 2\pi n\frac{q}{n}) \\
&= \sin(2\pi mt + 2\pi p) + \sin(2\pi nt + 2\pi q) \\
&= \sin(2\pi mt) + \sin(2\pi nt) = f(t).
\end{aligned}$$

Moreover, T is the period of f because for any τ, such that $f(t) = f(t+\tau)$, $1/\tau$ must be a common factor of m and n, however, $1/T$ is the greatest common divisor of m and n, so that $T \leq \tau$. □

2. *Gaussian random variables* Using the functions `randn` and `hist`, generate a sequence of N independent realizations of a zero mean, unit variance, Gaussian random variable, and plot the histogram. Repeat the experiment with $N = 100, 500, 2000, 10000$ realizations and comment on the results. Plot on top of the histogram the theoretical probability density function.

Solution:

```
N = [100,500,2000,10000];
ll = {'g','r','b','--k'};
```

```
x = randn(1,max(N));
b = linspace(-3,3,31);
figure, hold on
for i = 1:4
  h = hist(x(1:N(i)),b);
  h = h / max(h) / sqrt(2*pi);
  plot(b,h,ll{i}), % pause
  s{i} = sprintf('n = %d',N(i));
end
pdf = 1 / sqrt(2*pi) * exp( -b.^2/2 );
s{5} = 'theorerical';
plot(b,pdf,'-k'), legend(s)
```

The resulting plot is shown in Figure 9. □

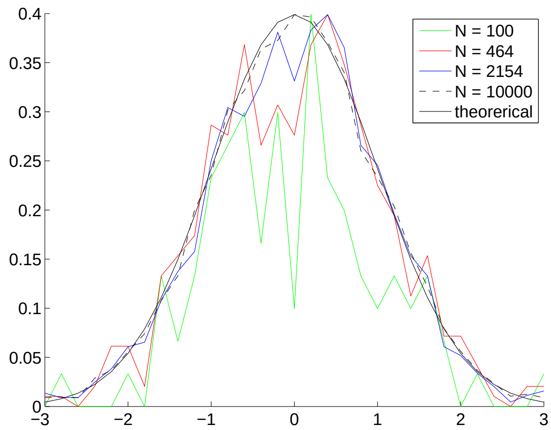

Figure 9: Empirical probability density functions of a sum of random variable for different number of realizations.

3. *Central limit theorem* In this exercise, you will experimentally confirm the central limit theorem, one version of which states that the distribution of the sum of independent random variables with arbitrary distributions tends to Gaussian as the number of variables tends to infinity. The distribution of the sum of two random variables is the convolution of the distributions of the two variables. Using this fact, the sum of n uniformly distributed random variables in the interval $[-0.5, 0.5]$ has distribution

$$p_n = \frac{1}{n} \underbrace{p \star \cdots \star p}_{n \text{ times}}.$$

Construct and plot p_n for $n = 1, 2, 3, 4, 5$.

Solution:

```
t = -3:0.01:3;
T = length(t); M = round(T/2);
p = zeros(1,T);
p(intersect(find(t >= -0.5),find(t <= 0.5))) = 1;
pn = p;
ll = {':','g','r','b','--k','-k'};
figure, hold on, plot(t,pn / max(pn) / sqrt(2*pi),ll{1})
s{1} = 'p'
for n = 2:5
  pn = conv(pn,p); pn = pn(M:end-M+1);
```

```
  plot(t,pn / max(pn) / sqrt(2*pi),ll{n});
  s{n} = [s{n-1} '*p'];
end
pdf = 1 / sqrt(2*pi) * exp( -t.^2/2 );
s{n+1} = 'theorerical';
plot(t,pdf,ll{n+1}), legend(s)
```

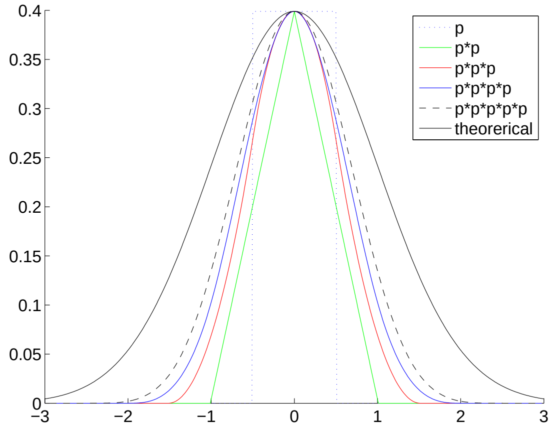

Figure 10: Successive convolutions of the rectangular pulse with itself approach the Gaussian distribution.

□

4. *Response of a linear time-invariant system to white Gaussian noise* Find the output of the filtered, defined by the transfer function

$$H(z) = \frac{1}{z - 0.5}$$

to white Gaussian noise and plot the correlation function of the output. Comment on the result. You can use Matlab's functions `randn`, `filter` and `xcorr`.

Solution:

```
T = 10000;
u = randn(1,T);
y = filter(1,[1 -0.5],u);
ry = xcorr(y);
M  = round(length(ry)/2); t = 15;
figure, plot(-t:t,ry(M-t:M+t)), grid on
xlabel('t'), ylabel('ry(t)')
```

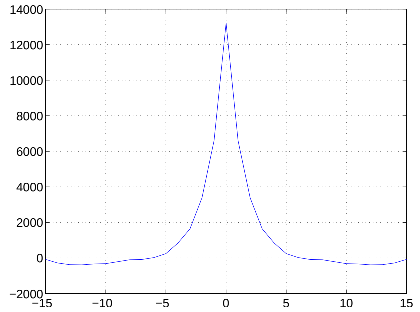

Figure 11: Correlation function of the output.

□