

ΕΣΠΑ 2007-2013, Ε.Π. «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ»
ΔΡΑΣΗ «ΑΡΙΣΤΕΙΑ II»

«Σύμφωνα Ασύρματα Οπτικά Συστήματα ως
Δομικά Στοιχεία για την Υλοποίηση Δικτύων
Οικίας και Επαγγελματικών Χώρων Επόμενης
Γενιάς, (COWS)»

Κωδικός Έργου : 3525

COWS

| | |
|---------------------|--|
| Παραδοτέο: | Π1.2 |
| Τίτλος: | Εργαλείο προσομοίωσης καναλιού και διαύλου |
| Ημερομηνία: | 31/1/2015 |
| Επιμέρους Δράση: | Π.Ε. 1 |
| Υπεύθυνος Σύνταξης: | Θωμάς Καμαλάκης |
| Ομάδα Εργασίας | Θωμάς Καμαλάκης, Κατερίνα Μαργαρίτη, Δαλάκας Βασίλειος, Βελάνας Παντελής |
| Είδος: | Τεχνική Αναφορά |



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

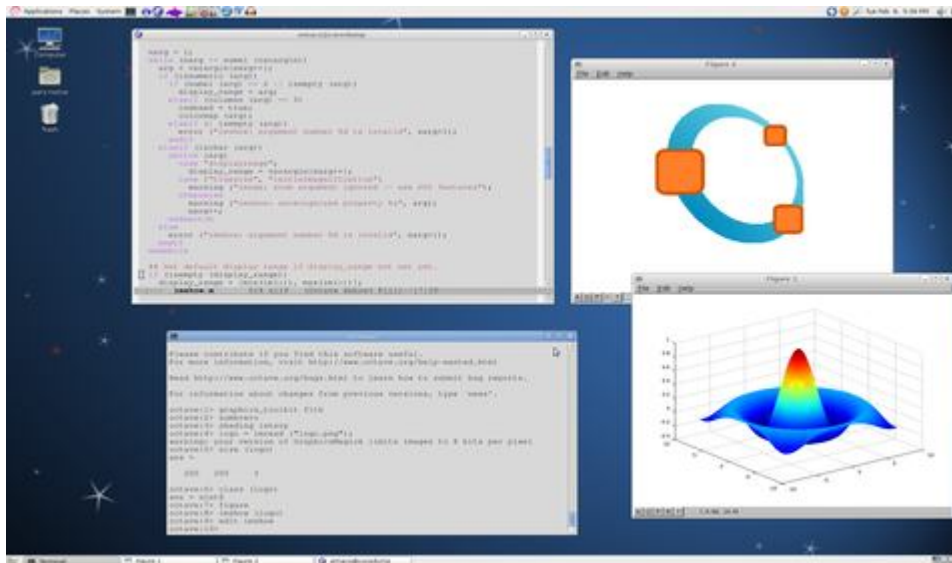


1. Περιεχόμενα

| | |
|---|----|
| 1. Περιεχόμενα..... | 2 |
| 2. Σκοπός του παραδοτέου..... | 4 |
| 3. Οπτικός Διαμορφωτής..... | 5 |
| 3.1 Αρχή Λειτουργίας..... | 5 |
| 3.2 Υλοποίηση..... | 6 |
| 4. Οπτικά Στοιχεία στον Πομπό..... | 13 |
| 4.1 Θεωρητικό Υπόβαθρο..... | 13 |
| 4.2 Υλοποίηση..... | 13 |
| 5. Υπολογισμός του Πεδίου..... | 16 |
| 5.1 Θεωρητικό Υπόβαθρο..... | 16 |
| 5.2 Υλοποίηση της διάδοσης της Gaussian δέσμης..... | 17 |
| 5.3 Gaussian δέσμες με διαφορετικό προσανατολισμό..... | 19 |
| 5.4 Επίδραση των οπτικών στοιχείων του πομπού..... | 24 |
| 6. Ισολογισμός Ισχύος..... | 31 |
| 6.1 Θεωρητικό υπόβαθρο..... | 31 |
| 6.2 Υλοποίηση..... | 31 |
| 7. Οπτικός Δέκτης..... | 34 |
| 7.1 Θεωρητικό Υπόβαθρο..... | 34 |
| 7.2 Υπολογισμός της ισχύος του προσθετικού θορύβου..... | 36 |
| 7.3 Υπολογισμός του προσθετικού θορύβου..... | 38 |
| 7.4 Ο θόρυβος φάσης..... | 41 |
| 7.5 Υπολογισμός των ρευμάτων εξόδου του δέκτη..... | 42 |
| 8. Αναφορές..... | 53 |

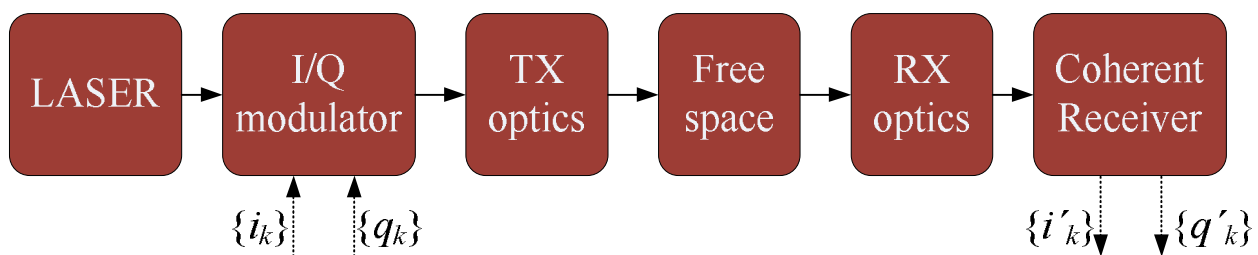
2. Σκοπός του παραδοτέου

Ο σκοπός του παρόντος παραδοτέου είναι να παρουσιάσει τον τρόπο υλοποίησης του μοντέλου του ασύρματου οπτικού καναλιού και της ασύρματης οπτικής ζεύξης με σύμφωνη φάραση. Το θεωρητικό υπόβαθρο του εν λόγω μοντέλου έχει σε μεγάλο βαθμό περιγραφεί στο Π1.1. κυρίως σε ότι αφορά το οπτικό ασύρματο κανάλι. Η υλοποίηση του μοντέλου έχει γίνει στο GNU-OCTAVE [1] το οποίο είναι ένα λογισμικό ελεύθερου κώδικα που αποτελεί συχνή επιλογή για την υλοποίηση τέτοιου είδους εργαλείων. Η συλλογή από octave files που μπορεί να χρησιμοποιηθεί για την προσομοίωση του συστήματος μπορεί να βρεθεί στον δικτυακό τόπο του προγράμματος. Στην Εικόνα 1 παρουσιάζουμε το γραφικό περιβάλλον του GNU-OCTAVE στο Debian Linux, ωστόσο πρέπει να αναφερθεί το εν λόγω λογισμικό είναι διαθέσιμο για διάφορα λειτουργικά συστήματα όπως Ubuntu, Fedora, Gentoo, SuSE, FreeBSD, OpenBSD, OS X ακόμα και Android.



Εικόνα 1: το περιβάλλον του GNU-OCTAVE

Στην Εικόνα 2 παρουσιάζουμε τις βασικές βαθμίδες του συστήματος όπως τις αναλύσαμε στο Π1.1 και αποτελείται από τον οπτικό πομπό (laser), ένα διαμορφωτή I/Q ο οποίος διαμορφώνει το οπτικό κύμα, τα οπτικά στοιχεία που πομπού (TX optics) το οποίο στη συνέχεια διαδίδεται στον ελεύθερο χώρο και προσπίπτει στα οπτικά στοιχεία του δέκτη (RX optics) και οδηγείται στο σύμφωνο δέκτη. Σκοπός μας είναι να συνδέσουμε τα σύμβολα $\{i_k, q_k\}$ στην είσοδο του διαμορφωτή I/Q με τις εκτιμήσεις των συμβόλων $\{i'_k, q'_k\}$.

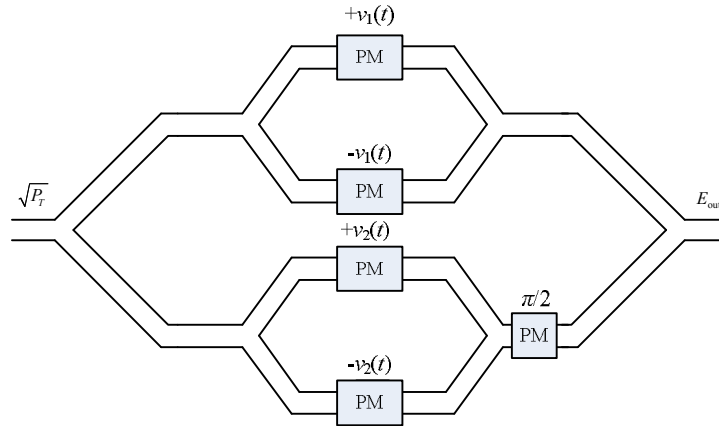


Εικόνα 2: Οι βαθμίδες του οπτικού ασύρματου συστήματος

Στην συνέχεια και βασιζόμενοι στο Π1.1 θα παρουσιάσουμε πώς υλοποιήσαμε τα διάφορα συστατικά του μοντέλου στο GNU-OCTAVE.

3. Οπτικός Διαμορφωτής

3.1 Αρχή Λειτουργίας



Εικόνα 3: Ένας οπτικός διαμορφωτής I/Q.

Ο διαμορφωτής είναι ίσως το πιο εύκολο προς μοντελοποίηση στοιχείο της ζεύξης. Στο Π1.1. παρουσιάσαμε την βασική μορφή ενός εμπορικά διαθέσιμου οπτικού διαμορφωτή I/Q την οποία και παραθέτουμε στην Εικόνα 3. Η εξίσωση που διέπει την έξοδο του διαμορφωτή είναι [2]:

$$E_{\text{out}} = \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{v_I}{V_\pi}\right) + j \frac{\sqrt{P_T}}{2} \cos\left(\pi \frac{v_Q}{V_\pi}\right) \quad (1)$$

όπου P_T είναι η ισχύς που εκπέμπεται από το LASER στον πομπό και v_1, v_2 είναι οι τάσεις που εφαρμόζουμε στα I και Q ηλεκτρόδια του διαμορφωτή αντίστοιχα. Η τάσεις αυτές εξαρτώνται από τα ψηφιακά σύμβολα i_k και q_k ως εξής:

$$v_I = \sum_{k=-\infty}^{+\infty} i_k p(t - kT) + V_b = v_i(t) + V_b \quad (2)$$

$$v_Q = \sum_{k=-\infty}^{+\infty} q_k p(t - kT) + V_b = v_q(t) + V_b \quad (3)$$

όπου $p(t)$ είναι ο παλμός που παράγεται από το ηλεκτρονικό κύκλωμα οδήγησης (π.χ. μία γεννήτρια αυθαίρετων σημάτων, κτλ) ενώ V_b είναι μία τάσης πόλωσης που προσθέτουμε στα σήματα τάσης. Συχνά ο παλμός αυτός προσεγγίζεται με έναν παλμό ανυψωμένου ημιτόνου,

$$p(t) = \begin{cases} 1 & , |t| < (1 - \beta) \frac{T}{2} \\ \frac{1}{2} \left[1 - \sin\left(\frac{\pi(|t| - \frac{T}{2})}{\beta T}\right) \right] & , |t| \geq (1 - \beta) \frac{T}{2} \text{ και } |t| \geq (1 + \beta) \frac{T}{2} \\ 0 & |t| \geq (1 + \beta) \frac{T}{2} \end{cases} \quad (4)$$

Στην Εικόνα 4 δείχνουμε μερικά παραδείγματα παλμών ανυψωμένου ημιτόνου για διάφορες τιμές της παραμέτρου β η οποία περιγράφει πόσο απότομα αποσβένει ο παλμός.

Θα πρέπει να αναφερθεί ότι ιδιαίτερη σημασία έχει η τάση πόλωσης V_b την οποία προσθέτουμε στα σήματα. Συχνά θέτουμε $V_b = \pm V_\pi/2$ που συχνά αναφέρεται και ως σημείο τετραγωνισμού (quadrature point). Στην περίπτωση όπου $V_b = -V_\pi/2$ τότε θα έχουμε

$$E_{\text{out}} = \frac{\sqrt{P_T}}{2} \sin\left(\pi \frac{v_I}{V_\pi}\right) + j \frac{\sqrt{P_T}}{2} \sin\left(\pi \frac{v_Q}{V_\pi}\right) \quad (5)$$

οπότε όσο οι τάσεις v_i και v_q παραμένουν εντός του διαστήματος $[-V_\pi/2, V_\pi/2]$ έπεται ότι το ηλεκτρικό πεδίο ακολουθεί τις μεταβολές της φάσης. Για $v_i, v_q \ll V_\pi/2$ έπεται ότι

$$E_{\text{out}} \cong \frac{\pi\sqrt{P_T}}{2V_\pi} \{v_i + jv_q\} \quad (6)$$

οπότε ο διαμορφωτής είναι σχεδόν γραμμικός.

3.2 Υλοποίηση

Ξεκινάμε με την υλοποίηση του παλμού ανυψωμένου ημιτόνου που υλοποιείται με την συνάρτηση *raised_cosine* η οποία φαίνεται και παρακάτω. Η συνάρτηση αυτή δημιουργεί έναν παλμό σύμφωνα με την (4) κεντραρισμένο στο $t=t_0$.

raised_cosine.m

```
function p=raised_cosine(t,t0,T,b);
% p=raised_cosine(t,t0,T,b);
%
% Realization of a raised cosine pulse
% t is the time axis
% t0 is the pulse center
% T is the symbol duration
% b is the roll-off factor
%
p=zeros(1,length(t));
t=t-t0;

t1=(1-b)*T/2;
t2=(1+b)*T/2;
i1=find( abs(t)<= t1 );
i2=find( (abs(t)>=t1) & (abs(t)<=t2) );

p(i1)=1;
p(i2)=1/2*(1-sin(pi*(abs(t(i2))-T/2)/b/T));
```

Για να δοκιμάσουμε την εν λόγω συνάρτηση χρησιμοποιούμε το αρχείο *raised_cosine_example* το οποίο δημιουργεί και παριστάνει γραφικά στους ίδιους άξονες τρεις παλμούς ανυψωμένου ημιτόνου με διαφορετική τιμή του β . Ο κώδικας του *raised_cosine_example* δίνεται παρακάτω:

raised_cosine_example.m

```
T=1e-9; % symbol duration

% example roll off factors
b1=0.01;
b2=0.5;
b3=1.0;

Nt=1000; % points in the t-axis
Tmax=2*T; % size of the time window
Dt=Tmax/Nt; % time step
t=(-Nt/2:1:(Nt/2-1))*Dt; % time axis

% Generation of raised cosine pulses
p1=raised_cosine(t,0,T,b1);
p2=raised_cosine(t,0,T,b2);
p3=raised_cosine(t,0,T,b3);

% Plot raised cosine pulses
plot(t/1e-9,p1,t/1e-9,p2,t/1e-9,p3,'LineWidth',2);
ylim([-0.1 1.1]);
set(gca,'FontSize',20);
legend('\beta=0.01','\beta=0.5','\beta=1');
ylabel('p(t)');
xlabel('t/T');
```

Στην Εικόνα 4 δείχνουμε το σχήμα που παράγει το αρχείο *raised_cosine_example*. Παρατηρούμε ότι για $\beta=0.01$ ο παλμός του ανυψωμένου ημιτόνου είναι ορθογώνιος ενώ όσο αυξάνει το β τόσο ο παλμός αποκλίνει από την ιδανική ορθογώνια του μορφή. Η εν λόγω συμπεριφορά είναι χαρακτηριστική γεννητριών αυθαίρετων κυματομορφών: όσο πλησιάζουμε στο μέγιστο αναλογικό εύρος ζώνης της γεννήτριας τόσο λιγότερο μπορεί να παρακολουθήσει η γεννήτρια τις μεταβολές των συμβόλων και οι παλμοί που παράγονται μοιάζουν με παλμούς ανυψωμένου συνημιτόνου με μεγάλο β . Στην βιβλιογραφία το β αναφέρεται συχνά ως παράγοντας μετάβασης (roll-off).

Έχει επίσης ενδιαφέρον να υπολογίσουμε το φάσμα των παλμών ανυψωμένου συνημιτόνου για διάφορες τιμές του β . Για το σκοπό αυτό χρησιμοποιούμε τον FFT του GNU-OCTAVE και το αρχείο *raised_cosine_spectra* του οποίου ο κώδικας περιγράφεται παρακάτω:

raised_cosine_spectra.m

```
close all
T=1e-9;           % symbol duration

% example roll off factors
b1=0.01;
b2=0.5;
b3=1.0;

Nt=16384;         % points in the t-axis
Tmax=50*T;       % size of the time window
Dt=Tmax/Nt;      % time step
t=(-Nt/2:1:(Nt/2-1))*Dt; % time axis

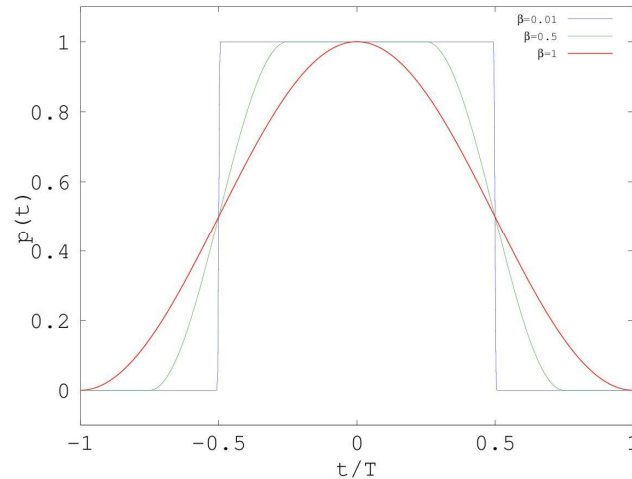
% Generation of raised cosine pulses
p1=raised_cosine(t,0,T,b1);
p2=raised_cosine(t,0,T,b2);
p3=raised_cosine(t,0,T,b3);

% Plot raised cosine pulses
plot(t/T,p1,t/T,p2,t/T,p3,'LineWidth',2);
xlim([-1 1]);
ylim([-0.1 1.1]);
set(gca,'FontName','Times New Roman','FontSize',16);
legend('\beta=0.01','\beta=0.5','\beta=1');
ylabel('p(t)');
xlabel('t/T');

% Estimate Fourier transform through FFT
Df=1/Tmax;       % Frequency step
f=(-Nt/2:1:(Nt/2-1))*Df; % Frequency axis

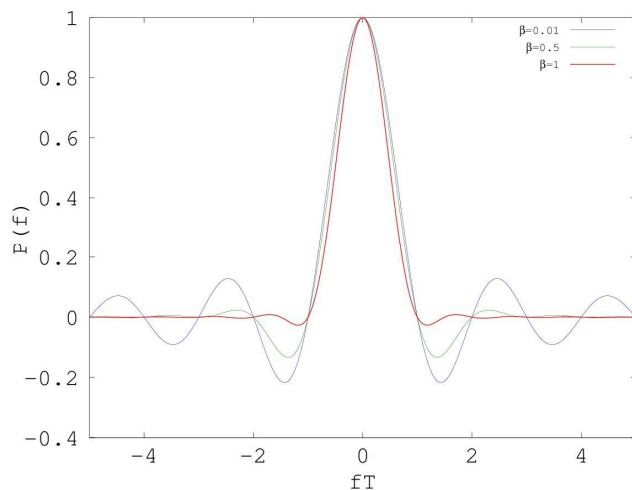
% Pulse spectra
P1=Dt*fftshift(fft(fftshift(p1)));
P2=Dt*fftshift(fft(fftshift(p2)));
P3=Dt*fftshift(fft(fftshift(p3)));

% Plot raised cosine spectra
figure(2);
plot(f*T,P1/max(P1),f*T,P2/max(P2),f*T,P3/max(P3),'LineWidth',2);
set(gca,'FontSize',20);
legend('\beta=0.01','\beta=0.5','\beta=1');
ylabel('P(f)');
xlabel('fT');
xlim([-5 5]);
```



Εικόνα 4: Παραδείγματα παλμών ανυψωμένου ημιτόνου.

Στην Εικόνα 5 έχουμε παραστήσει γραφικά το κανονικοποιημένο φάσμα των παλμών ανυψωμένου ημιτόνου που υπολογίζονται από το αρχείο *raised_cosine_spectra*. Όπως αναμέναμε όσο αυξάνει η τιμή του β , τόσο μεγαλύτερη είναι η καταστολή των πλευρικών λοβών του φάσματος και επομένως τόσο περισσότερο αποκλίνει η κυματομορφή από τον ιδανικό ορθογώνιο παλμό.



Εικόνα 5: Το φάσμα των παλμών ανυψωμένου ημιτόνου.

Το επόμενο βήμα είναι να υλοποιήσουμε μία συνάρτηση που να υλοποιεί τα σήματα τάσης που οδηγούν τον διαμορφωτή και δίνονται από τις (2) και (3). Η συγκεκριμένη λειτουργία γίνεται από την συνάρτηση *driving_voltage* της οποίας τον κώδικα και παραθέτουμε:

driving_voltage.m

```
function v=driving_voltage(t,sm,T,b);
%
% v=driving_voltage(t,sm,T,b);
%
% returns the driving voltage corresponding to the digital symbols
% amplitudes sm. Note that sm are taken real.
% T is the symbol period
% t is the time axis
% b is the roll off parameter of the raised cosine pulse
% the first symbol occupies the interval [-T/2 T/2], the second symbol [T/2
% 3*T/2] and so on

v=zeros(1,length(t));
```



```

for m=1:length(sm)
    tm=m*T;
    v=v+sm(m)*raised_cosine(t,tm,T,b);
end

```

pulsetrain_example.m

```

T=1e-9;           % symbol duration

% example roll off factors
b1=0.01;
b2=0.5;
b3=1.0;

sm=[1 0 1 0 1 1 0 0];           % symbol amplitudes
Nt=50;                          % points in the t-axis / symbol
Ntotal=(length(sm)+2)*Nt;       % total number of symbols in the t-axis
Tmin=-T;                         % start of the time window
Tmax=Tmin+(length(sm)+2)*T;     % end of the time window
Dt=(Tmax-Tmin)/Ntotal;          % time step
t=Tmin+(1:Ntotal)*Dt;           % time axis

% Generation of driving voltages
p1=driving_voltage(t,sm,T,b1);
p2=driving_voltage(t,sm,T,b2);
p3=driving_voltage(t,sm,T,b3);

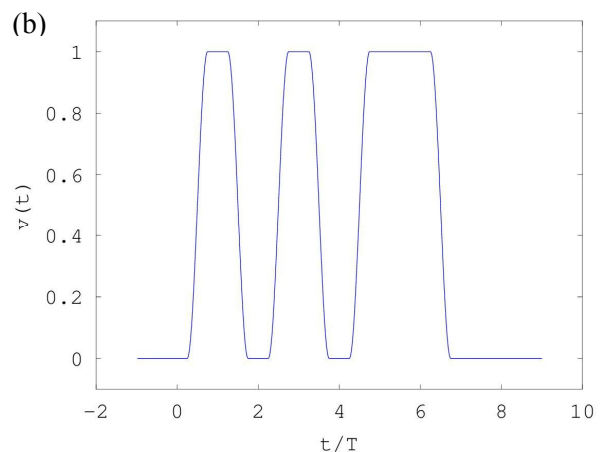
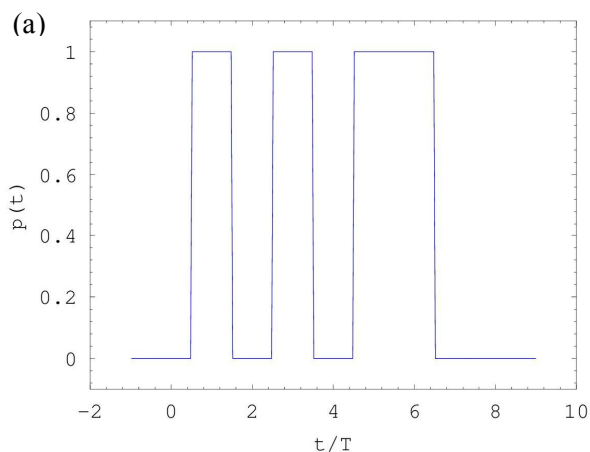
% Plot raised cosine pulses

figure(1);
plot(t/1e-9,p1,'LineWidth',2);
set(gca,'FontSize',20);
ylim([-0.1 1.1]);
ylabel('p(t)');
xlabel('t/T');

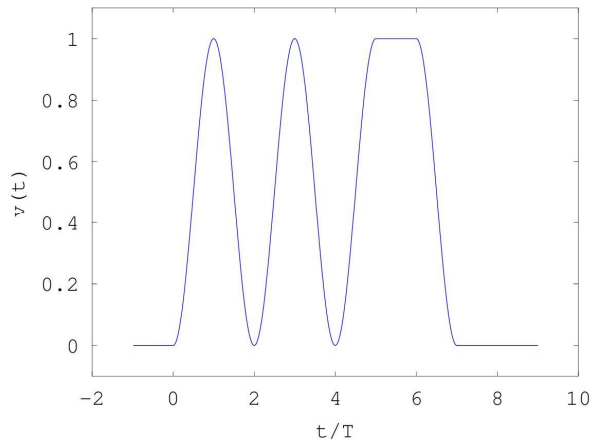
figure(2);
plot(t/1e-9,p2,'LineWidth',2);
set(gca,'FontSize',20);
ylim([-0.1 1.1]);
ylabel('v(t)');
xlabel('t/T');

figure(3);
plot(t/1e-9,p3,'LineWidth',2);
set(gca,'FontSize',20);
ylim([-0.1 1.1]);
ylabel('v(t)');
xlabel('t/T');

```



(c)



Εικόνα 6: Η παλμοσειρά που αντιστοιχεί στα διαδοχικά πλάτη συμβόλων 1 0 1 0 1 1 0 0 στην περίπτωση όπου a) $\beta=0.01$, b) $\beta=0.5$ και c) $\beta=1$.

Το αρχείο OCTAVE *pulsetrain_example* παρουσιάζει ένα παράδειγμα χρήσης της συνάρτησης *driving_voltage*. Η μεταβλητή *sm* περιέχει τα πλάτη των διαδοχικών συμβόλων. Στην Εικόνα 6 δείχνουμε τις παλμοσειρές που υπολογίζονται από την *pulsetrain_example* στην περίπτωση όπου τα πλάτη των συμβόλων είναι 1 0 1 0 1 1 0 0.

Το τελικό βήμα είναι να υλοποιήσουμε τον διαμορφωτή I/Q βάσει της εξίσωσης (4). Η σχετική συνάρτηση *iq_modulator* είναι πολύ απλή και την παραθέτουμε παρακάτω μαζί με το αρχείο *iqmodulation_example* το οποίο δείχνει ένα παράδειγμα χρήσης της συγκεκριμένης συνάρτησης.

iq_modulator.m

```
function E=iq_modulator(t,vI,vQ,Vpi,PT);
%
% p=iq_modulator(t,v1,v2,Vpi,PT);
%
% Returns the baseband output of the IQ modulator
% vI is the voltage driving the modulator I electrode
% vQ is the voltage driving the modulator Q electrode
% Vpi is the modulator characteristic voltage
% PT is the input power

E=sqrt(PT)/2*cos(pi*vI/Vpi) + j*sqrt(PT)/2*cos(pi*vQ/Vpi);
```

iq_modulation_example.m

```
PT=1e-3; % input power
T=1e-9; % symbol duration
b=0.5; % Roll off factor
Vpi=1; % modulator characteristic voltage
smI=[1 0 1 0 1 1 0 0]; % symbol amplitudes for I voltage component
smQ=[1 0 1 0 1 1 0 0]; % symbol amplitudes for Q voltage component
Nt=50; % points in the t-axis / symbol
Ntotal=(length(smQ)+2)*Nt; % total number of symbols in the t-axis
Tmin=-T; % start of the time window
Tmax=Tmin+(length(smQ)+2)*T; % end of the time window
Dt=(Tmax-Tmin)/Ntotal; % time step
t=Tmin+(1:Ntotal)*Dt; % time axis

% Generation of driving voltages
vI=driving_voltage(t,smI,T,b);
vQ=driving_voltage(t,smQ,T,b);

% Estimation of optical baseband signal
E=iq_modulator(t,vI,vQ,Vpi,PT);

% Plot raised cosine pulses

figure(1);
```

```

plot(t/1e-9,vI,'LineWidth',2);
ylim([-0.1 1.1]);
set(gca,'FontSize',20);
ylabel('v\rm_I(t\rm)');
xlabel('t /T');

figure(2);
plot(t/1e-9,vQ,'LineWidth',2);
ylim([-0.1 1.1]);
set(gca,'FontSize',20);
ylabel('v_Q(t)');
xlabel('t /T');

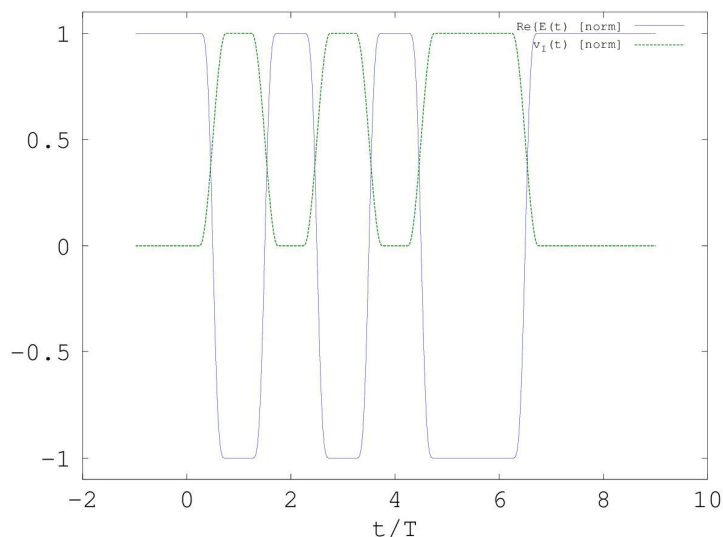
figure(3);
plot(t/1e-9,real(E),'LineWidth',2);
ylim([-1.1*max(real(E)) 1.1*max(real(E))]);
set(gca,'FontSize',20);
ylabel('Re\{E(t)\}');
xlabel('t /T');

figure(4);
plot(t/1e-9,imag(E),'LineWidth',2);
ylim([-1.1*max(imag(E)) 1.1*max(imag(E))]);
set(gca,'FontSize',20);
ylabel('Im\{E(t)\}');
xlabel('t /T');

figure(5);
plot(t/1e-9,real(E)/max(real(E)),t/1e-9,vI/max(vI),'--','LineWidth',2);
ylim([-1.1 1.1]);
set(gca,'FontSize',20);
xlabel('t /T');
legend('Re\{E(t)\} [norm]','v_I(t) [norm]');

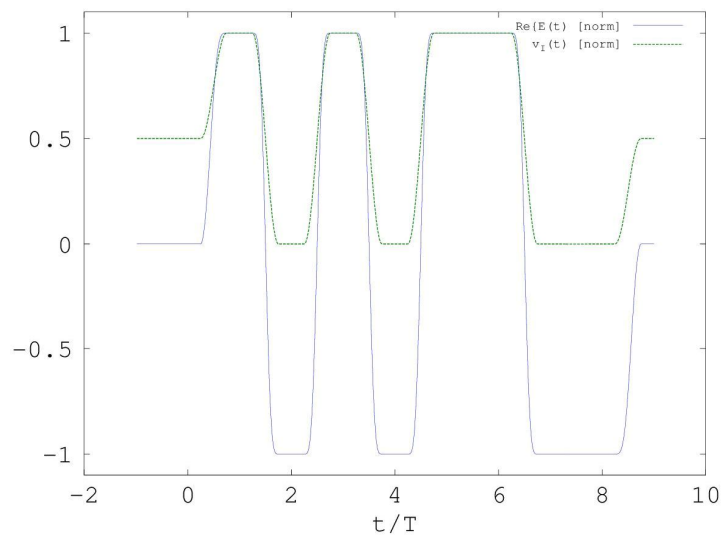
```

Ιδιαίτερο ενδιαφέρον παρουσιάζει η σύγκριση των ηλεκτρονικών κυματομορφών, π.χ. την $v_I(t)$ και την κυματομορφή του ηλεκτρικού πεδίου όπως αυτή αποτυπώνεται στο πραγματικό μέρος του. Στην Εικόνα 7 δείχνουμε την σύγκριση αυτή όπου παρατηρούμε ότι οι μικρές τάσεις μεταφράζονται σε υψηλό θετικό πεδίο ενώ οι υψηλές τάσεις σε χαμηλές (αρνητικές) τάσεις. Τονίζεται πως έχουμε επιλέξει το πλάτος της τάσης να είναι κοντά στην τάση $V_\pi=1V$ όπως φαίνεται και από τον κώδικα της `iqmodulation_example`.



Εικόνα 7: Σύγκριση των κυματομορφών $v_I(t)$ και $\text{Re}\{E(t)\}$.

Στην επόμενη εικόνα δείχνουμε την ίδια σύγκριση για την περίπτωση όπου το πλάτος των τάσης είναι ίσο με $V_\pi/2$, τα πλάτη των συμβόλων είναι 1 -1 1 -1 1 1 -1 -1 ενώ η τάση πόλωσης είναι $V_b=-V_\pi/2$. Όπως εξηγήσαμε και στην προηγούμενη παράγραφο στην περίπτωση αυτή οι μεταβολές του πεδίου ακολουθούνε πιο πιστά τις μεταβολές της τάσης οδήγησης.



Εικόνα 8: Σύγκριση των κυματομορφών $v_1(t)$ και $\text{Re}\{E(t)\}$ για $V_b = V_\pi/2$.

4. Οπτικά Στοιχεία στον Πομπό

4.1 Θεωρητικό Υπόβαθρο

Στην παρούσα παράγραφο θα ασχοληθούμε με την μοντελοποίηση των οπτικών στοιχείων του δέκτη. Όπως είδαμε στο Π1.1 [2] οι δέσμη που εκπέμπεται από το LASER ή τον διαμορφωτή I/Q μπορούν να θεωρηθούν Gaussian και τόσο η διάδοση τους στον ελεύθερο χώρο όσο και από συστήματα φακών περιγράφονται βάσει της θεωρίας των πινάκων ABCD [4]. Μία Gaussian δέσμη περιγράφεται από τον μιγαδικό παράγοντα q όπου:

$$\frac{1}{q} = -j \frac{\lambda}{\pi w^2} + \frac{1}{R} \quad (7)$$

Στην (7) με w συμβολίζουμε το $1/e^2$ εύρος της δέσμης και με R την ακτίνα καμπυλότητας της. Η Gaussian δέσμη δίνεται από την εξίσωση:

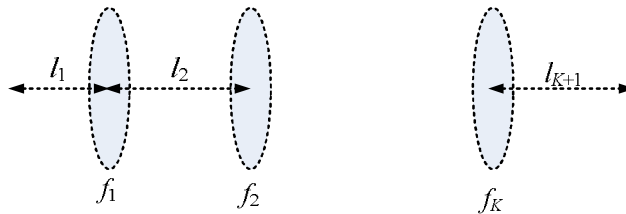
$$u_0(x, y) \sim \exp\left(-\frac{x^2 + y^2}{w^2} - jk \left[\frac{x^2 + y^2}{2R}\right]\right) \quad (8)$$

Για να περιγράψουμε την διάδοση μίας Gaussian δέσμης σε ένα οπτικό σύστημα υπολογίζουμε τον συνολικό ABCD πίνακα του συστήματος ως το γινόμενο των επιμέρους πινάκων των δομικών του στοιχείων. Ο ABCD πίνακας του ελεύθερου χώρου δίνεται από την σχέση,

$$M_{FP}(d) = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \quad (9)$$

όπου d είναι η απόσταση διάδοσης ενώ στην περίπτωση ενός φακού με εστιακή απόσταση f ο σχετικός πίνακας δίνεται από την σχέση:

$$M_L(f) = \begin{bmatrix} 1 & 0 \\ -1/f & 1 \end{bmatrix} \quad (10)$$



Εικόνα 9: Παράδειγμα οπτικού συστήματος.

Στην Εικόνα 9 παρουσιάζουμε την δομή ενός οπτικού συστήματος πολλαπλών φακών όπου l_i είναι οι αποστάσεις μεταξύ των διαδοχικών φακών και f_i είναι οι εστιακές αποστάσεις των φακών. Ο συνολικός πίνακας ABCD δίνεται από την σχέση:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = M_{FP}(l_1) M_L(f_1) \cdots M_{FP}(l_K) M_L(f_K) M_{FP}(l_{K+1}) \quad (11)$$

Αν θεωρήσουμε ότι ο παράγοντας q είναι ίσος με q_i στην είσοδο του συστήματος και με q_o στην έξοδο τότε θα έχουμε:

$$q_o = \frac{Aq_i + B}{Cq_i + D} \quad (12)$$

4.2 Υλοποίηση

Για να περιγράψουμε την διάδοση της δέσμης από ένα γενικό οπτικό σύστημα φακών χρησιμοποιούμε την συνάρτηση *propagation_lens* την οποία παρουσιάζουμε παρακάτω:

propagation_lens.m

```

function [q,w,R]=propagation_lens(li,fi,w0,R0,lambda)
%
% [q,w,R]=propagation_lens(li,fi,w0,R0,lambda);
%
% Propagation of Gaussian beam through an optical system composed of a
% series of lenses.
% li(2:n) are the distances between the lenses
% li(1) is the distance of the source and the first lens
% fi(1:n-1) are the focal lengths of the lenses
% w0 is the original 1/e^2 width of the beam at the source
% R0 is the initial radius of curvature (should be set to Inf if the beam
% is transmitted from the laser source)

% initial q parameter calculation
q0inv=1/R0-j*lambda/pi/w0^2;
q0=1/q0inv;
M=eye(2,2);

for m=1:length(fi)

    % ABCD matrix for free space propagation
    M1=[1 li(m); 0 1];

    % ABCD matrix for lens
    M2=[1 0; -1/fi(m) 1];

    % update ensemble ABCD matrix
    M=M1*M2*M;
end

% Isolate matrix elements
A=M(1,1);
B=M(1,2);
C=M(2,1);
D=M(2,2);

% Calculate output q factor
q=(A*q0+B)/(C*q0+D);

qr=real(1/q);
qi=imag(1/q);

%Estimate beam width and beam radius
R=1/qr;
w=sqrt(-lambda/pi/qi);

```

Το αρχείο *TX_propagation_example* δείχνει ένα παράδειγμα χρήσης της συνάρτησης *propagation_lens* στο οποίο συγκρίνουμε τα χαρακτηριστικά της δέσμης στον ελεύθερο χώρο και στην περίπτωση όπου η δέσμη περνάει από ένα φακό με εστιακή απόσταση $f=10\text{mm}$ ο οποίος απέχει απόσταση $d=f$ από την πηγή εκπομπής. Θεωρούμε ότι η αρχική δέσμη έχει εύρος $w=1\mu\text{m}$. Στην Εικόνα 10 δείχνουμε την εικόνα που παράγει η *TX_propagation_example*. Παρατηρούμε ότι στην δεύτερη περίπτωση το εύρος της δέσμης παραμένει σταθερή επειδή έχουμε $d=f$ και επομένως παράγεται εστιασμένη δέσμη.

TX_propagation_example.m

```

close all;

wi=1e-6;           % Initial beamwidth
Ri=Inf;           % initial radius of curvature
lambda=780e-9;    % Wavelength
f=10e-3;          % Focal length of lenses
d1=f;             % distance from first lens
L=10.^(-6:0.01:1); % distances assumed from the second lens.

% calculate beam parameters at various distances

```

```

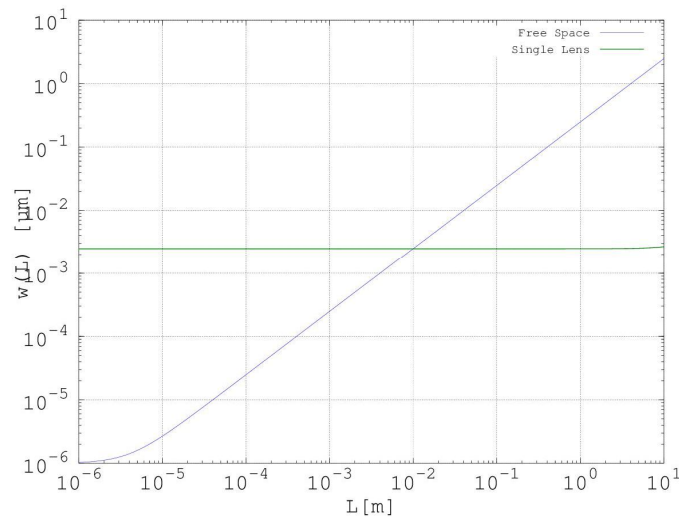
w0=zeros(1,length(L));
R0=zeros(1,length(L));
w2=zeros(1,length(L));
R2=zeros(1,length(L));

% first assume only free space propagation
fi=[];
for m=1:length(L)
    li=L(m);
    [q1,w0(m),R0(m),M1]=propagation_lens(li,fi,wi,Ri,lambda);
end

% second assume a single lens system
fi=[f];
for m=1:length(L)
    li=[d1 L(m)];
    [q2,w2(m),R2(m),M]=propagation_lens(li,fi,wi,Ri,lambda);
end

% Plot distance
figure(1);
loglog(L,w0,L,w2,'LineWidth',2);
grid
set(gca,'FontSize',20);
ylabel('w(L) [um]');
xlabel('L[m]');
legend('Free Space','Single Lens');

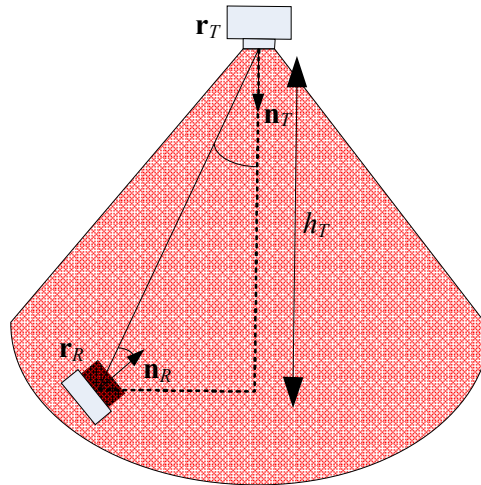
```



Εικόνα 10: Σύγκριση του εύρους της δέσμης στην περίπτωση όπου η δέσμη διαδίδεται στον ελεύθερο χώρο ή περνάει από ένα φακό με εστιακή απόσταση $f=10\text{mm}$ ο οποίος βρίσκεται σε απόσταση $d=f$ από την πηγή.

5. Υπολογισμός του Πεδίου

5.1 Θεωρητικό Υπόβαθρο



Εικόνα 11: Βοηθητικό σχήμα για τον ισολογισμό ισχύος της ζεύξης

Στο Π1.1 είδαμε το βασικό μαθηματικό μοντέλο που μπορεί να χρησιμοποιηθεί για τον ισολογισμό ισχύος της ζεύξης [2]. Στην Εικόνα 11 παρουσιάζουμε ένα βοηθητικό σχήμα που θα μας βοηθήσει για τον ισολογισμό ισχύος. Στο παρακάτω σχήμα έχουμε σημειώσει τις παρακάτω παραμέτρους

- Η θέση του πομπού \mathbf{r}_T στον τρισδιάστατο χώρο
- Η θέση του πομπού \mathbf{r}_R στον τρισδιάστατο χώρο
- Την κάθετη απόσταση μεταξύ πομπού και δέκτη L .
- Τον προσανατολισμό του πομπού όπως εκφράζεται από το διάνυσμα \mathbf{n}_T που δείχνει την κατεύθυνση μέγιστης διάδοσης.
- Τον προσανατολισμό του δέκτη όπως εκφράζεται από το διάνυσμα \mathbf{n}_R που δείχνει την κατεύθυνση μέγιστης λήψης.

Για ευκολία μπορούμε να θεωρήσουμε ότι $\mathbf{n}_T = -\mathbf{z} = (0, 0, -1)^t$ δηλαδή ότι ο πομπός εκπέμπει από ένα ύψος προς τα κάτω και πως $\mathbf{r}_T = (0, 0, h_T)$ ενώ ο δέκτης βρίσκεται στο επίπεδο $z = z_R$ και επομένως $\mathbf{r}_R = [x_R, y_R, z_R]^t$.

Στο επίπεδο του δέκτη η οπτική δέσμη δίνεται από την εξίσωση [2]:

$$u_0(x, y, z) = \sqrt{\frac{2P_T}{\pi w_x(z)w_y(z)}} \exp\left(-\frac{x^2}{w_x^2(z)} - \frac{y^2}{w_y^2(z)} - jk\left[\frac{x^2}{2R_x(z)} + \frac{y^2}{2R_y(z)}\right] + j\phi(z)\right) \quad (13)$$

όπου

$$w_x^2(z) = w_x^2 + \left[\frac{z}{2kw_x}\right]^2 = w_x^2 \left\{1 + \left[\frac{z}{z_{Rx}}\right]^2\right\} \quad (14)$$

$$w_y^2(z) = w_y^2 + \left[\frac{z}{2kw_y}\right]^2 = w_y^2 \left\{1 + \left[\frac{z}{z_{Ry}}\right]^2\right\} \quad (15)$$

$$R_x(z) = z \left\{1 + \left(\frac{z_{Rx}}{z}\right)^2\right\} \quad (16)$$

$$R_y(z) = z \left\{ 1 + \left(\frac{z_{Ry}}{z} \right)^2 \right\} \quad (17)$$

$$\phi(z) = \frac{1}{2} \operatorname{atan} \left(\frac{z}{z_{Rx}} \right) + \frac{1}{2} \operatorname{atan} \left(\frac{z}{z_{Ry}} \right) \quad (18)$$

Οι παράμετροι z_{Rx} και z_{Ry} είναι τα μήκη Rayleigh και δίνονται από τις παρακάτω σχέσεις [2]:

$$z_{Rx} = \frac{\pi w_x^2}{\lambda} \quad (19)$$

$$z_{Ry} = \frac{\pi w_y^2}{\lambda} \quad (20)$$

Η πυκνότητα ισχύος υπολογίζεται από την σχέση:

$$\mathbf{W} = |u_0|^2 \mathbf{k} = \frac{2P_T}{\pi w_x(z) w_y(z)} \exp \left(-2 \frac{x^2}{w_x^2(z)} - 2 \frac{y^2}{w_y^2(z)} \right) \quad (21)$$

ενώ η οπτική ισχύς που προσπίπτει στην επιφάνεια του δέκτη είναι,

$$P_R \cong \frac{2P_T A_R \cos \theta_R}{\pi w^2(z)} \exp \left(-2 \frac{x^2 + y^2}{w^2(z)} \right) \quad (22)$$

όπου A_R είναι η επιφάνεια του δέκτη και

$$\cos \theta_R = \frac{\mathbf{n}_R \cdot (\mathbf{r}_T - \mathbf{r}_R)}{|\mathbf{r}_T - \mathbf{r}_R|} \quad (23)$$

5.2 Υλοποίηση της διάδοσης της Gaussian δέσμης

Για τον υπολογισμό της εξέλιξης της Gaussian δέσμης έχουμε υλοποιήσει το αρχείο *gaussian_beam* ενώ ένα παράδειγμα χρήσης του έχει υλοποιηθεί στο αρχείο *gaussian_beam_example*. Ο σχετικός κώδικας των δύο αρχείων παρατίθεται παρακάτω:

gaussian_beam.m

```
function [u,U]=gaussian_beam(PT,xR,yR,zR,w0x,w0y,lambda)
% function [u,U,phi,phi_z]=gaussian_beam(PT,xR,yR,L,w0x,w0y)
%
% Gaussian beam evolution estimation. Source is assumed to be located at
% (0,0,0) and the propagation is assumed along the z-axis
% PT is the transmitted power
% xR, yR, zR are the points in the 3D space where the Gaussian field is
% estimated.
% w0x is the initial 1/e^2 along the x-axis
% w0y is the initial 1/e^2 along the y-axis
% u is the Gaussian beam field
% U is the Gaussian beam intensity |u|^2

k=2*pi/lambda;
zRx=pi*w0x^2/lambda;
zRy=pi*w0y^2/lambda;
wxz=w0x*sqrt(1+(zR./zRx).^2);
wyz=w0y*sqrt(1+(zR./zRy).^2);
Rxz=zR.*(1+(zRx./zR).^2);
Ryz=zR.*(1+(zRy./zR).^2);
amp=sqrt(2*PT./wxz./wyz/pi);
phi_z=1/2*atan(zR./zRx)+1/2*atan(zR./zRy);
u0=amp.*exp(-xR.^2./wxz.^2-yR.^2./wyz.^2);
phi=-k*(xR.^2/2./Rxz+yR.^2/2./Ryz)+phi_z;
u=u0.*exp(j*phi);

% propagation constant
% Rayleigh length along x
% Rayleigh length along y
% beam width along x at z
% beam width along y at z
% radius of curvature along x
% radius of curvature along y
% maximum amplitude of Gaussian beam
% z-dependent phase increment
% amplitude of the beam
% phase of the beam
% Gaussian beam field
```

```

U=abs(u).^2; % Gaussian beam intensity

gaussian_beam_example.m

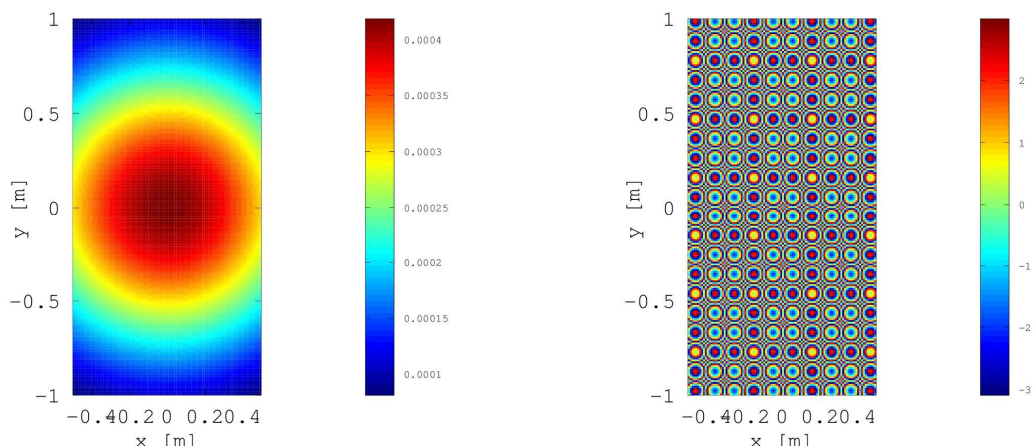
% Gaussian beam propagation example
close all;

w0x=2e-6; % Beam width along x
w0y=2e-6; % Beam width along y
L=5; % propagation distance
lambda=1.55e-6; % free space wavelength
PT=1e-3; % transmission power
rmin=-0.5;
rmax=0.5;
Nr=100;
x=rmin:(rmax-rmin)/Nr:rmax; % coordinates on the receiver plane
along the x-axis
y=2*rmin:(rmax-rmin)/Nr:2*rmax; % coordinates on the receiver plane
along the y-axis
[xR,yR]=meshgrid(x,y); % receiver plane x,y-coordinates
zR=L*ones(size(xR)); % receiver plane z-coordinates
[u,U]=gaussian_beam(PT,xR,yR,zR,w0x,w0y,lambda); % estimation of the Gaussian field

% Plot gaussian beam intensity
figure(1);
pcolor(xR,yR,U);
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

% Plot gaussian beam angle
figure(2);
pcolor(xR,yR,angle(u));
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

```

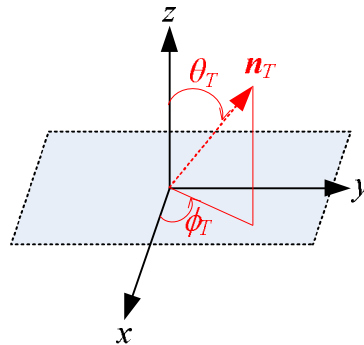


Εικόνα 12: Η ένταση και η φάση της Gaussian δέσμης του παραδείγματος.

Στην Εικόνα 12 δείχνουμε την ένταση και την φάση της Gaussian δέσμης όπως αυτή υπολογίζεται στο παράδειγμα *gaussian_beam_example*.

5.3 Gaussian δέσμες με διαφορετικό προσανατολισμό

Στη συνέχεια προχωρούμε στην υλοποίηση της εξέλιξης της Gaussian δέσμης με τυχαίο προσανατολισμό. Μπορούμε να χρησιμοποιήσουμε την προηγούμενη υλοποίηση όπου έχουμε θεωρήσει ότι την ειδική περίπτωση όπου η διεύθυνση διάδοσης είναι $\mathbf{n}_T=(0,0,1)$ με κατάλληλες στροφές αξόνων.



Εικόνα 13: Ένα γενικότερο σχήμα για τον υπολογισμό της Gaussian δέσμης στην περίπτωση διαφορετικής διεύθυνσης διάδοσης.

Στην Εικόνα 13 δείχνουμε την περίπτωση όπου το διάνυσμα της διεύθυνσης διάδοσης έχει διαφορετικό προσανατολισμό. Σκοπός μας είναι με μία κατάλληλη στροφή αξόνων να υπολογίσουμε την Gaussian δέσμη που διαδίδεται προς την νέα διεύθυνση \mathbf{n}_T . Για να συμπέσει το διάνυσμα \mathbf{n}_T με το διάνυσμα \mathbf{z} πρέπει να στρέψουμε τον άξονα των y στην κατεύθυνση της φοράς του ρολογιού κατά την γωνία $-\theta_T$ και στην συνέχεια να κάνουμε στροφή του άξονα των z δεξιόστροφα κατά την γωνία $-\phi_T$. Επομένως αν $u_0(x,y,z)$ είναι η εξέλιξη της Gaussian δέσμης στην πρώτη περίπτωση, η αντίστοιχη δέσμη που διαδίδεται ως προς την διεύθυνση \mathbf{n}_T είναι η $u_0(X,Y,Z)$ όπου

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R}_z(-\phi_T)\mathbf{R}_y(-\theta_T) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (24)$$

όπου οι πίνακες \mathbf{R}_y και \mathbf{R}_z είναι οι πίνακες δεξιόστροφης στροφής των αξόνων y και z . Επίσης αν η πηγή μας βρίσκεται στην θέση x_0,y_0,z_0 τότε αντί της $u_0(x,y,z)$ θεωρούμε την $u_0(x-x_0,y-y_0,z-z_0)$ και μετά κάνουμε την περιστροφή στην (24).

Για να υπολογίσουμε επομένως την δέσμη Gauss στην γενικότερη περίπτωση αρχικά πρέπει να υλοποιήσουμε τους πίνακες στροφής \mathbf{R}_y και \mathbf{R}_z . Αποφασίσαμε να υλοποιήσουμε και τον \mathbf{R}_x αν και δεν χρειάζεται στην περίπτωση μας, για λόγους πληρότητας. Οι αντίστοιχες συναρτήσεις είναι οι *rotmx*, *rotmy*, *rotmz*.

rotmx.m

```
function Rx=rotmx(theta);
% rotation matrix around the x-axis

Rx=[1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)];
```

rotmy.m

```
function Ry=rotmy(theta);

% rotation matrix around the y-axis
Ry=[cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)];
```

rotmz.m

```
function Rz=rotmz(theta);
% rotation matrix around the z-axis

Rz=[cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1];
```

Στη συνέχεια χρησιμοποιώντας τους πίνακες στροφής, υλοποιούμε τις συναρτήσεις που στρέφουν τους άξονες και υπολογίζουν τις συντεταγμένες των σημείων της ακτίνας. Πρόκειται για τις συναρτήσεις *rotation_x*, *rotation_y* και *rotation_z*.

rotation_x.m

```
function [xf yf zf]=rotation_y(xi,yi,zi,theta);
% function rf=rotation_y(ri);
%
% Estimates the coordinates of a given set of points when the y-axis is
% rotated by an angle theta clockwise by an angle theta
% (xi,yi,zi) are the coordinates of the points in the old coordinate system
% (xf,yf,zf) are the coordinates of the points in the new coordinate system
%
% rotation matrix
Rx=[1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)];

% rearrange points in a 3 x N matrix
a=size(xi);
ntot=prod(a);
xii=reshape(xi,1,ntot);
yii=reshape(yi,1,ntot);
zii=reshape(zi,1,ntot);
ri=[xii; yii; zii];
% rotate points

rf=Rx*ri;
% rearrange points in original dimensions
xff=rf(1,:);
yff=rf(2,:);
zff=rf(3,:);

xf=reshape(xff,a);
yf=reshape(yff,a);
zf=reshape(zff,a);
```

rotation_y.m

```
function [xf yf zf]=rotation_y(xi,yi,zi,theta);
% function rf=rotation_y(ri);
%
% Estimates the coordinates of a given set of points when the y-axis is
% rotated by an angle theta clockwise by an angle theta
% (xi,yi,zi) are the coordinates of the points in the old coordinate system
% (xf,yf,zf) are the coordinates of the points in the new coordinate system
%
% rotation matrix
Ry=[cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)];

% rearrange points in a 3 x N matrix
a=size(xi);
ntot=prod(a);
xii=reshape(xi,1,ntot);
yii=reshape(yi,1,ntot);
zii=reshape(zi,1,ntot);
ri=[xii; yii; zii];
% rotate points

rf=Ry*ri;
% rearrange points in original dimensions
xff=rf(1,:);
yff=rf(2,:);
zff=rf(3,:);

xf=reshape(xff,a);
yf=reshape(yff,a);
zf=reshape(zff,a);
```

```

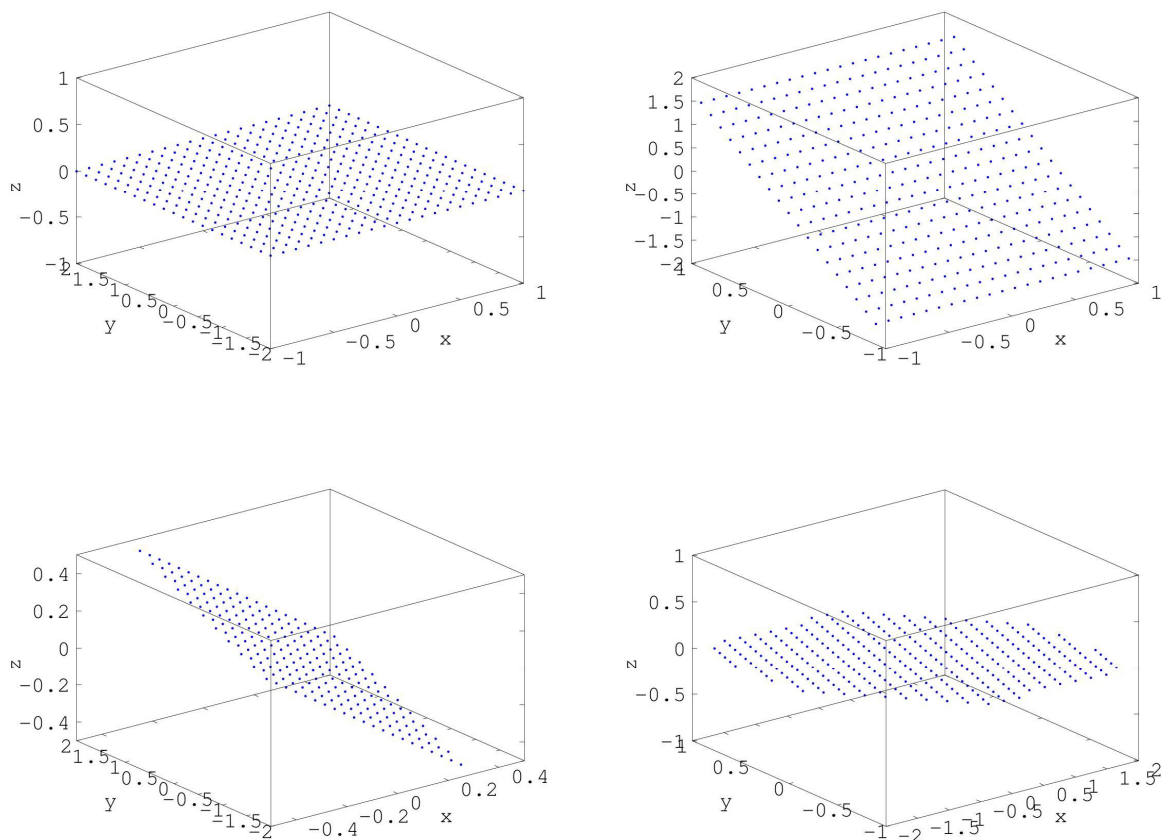
function [xf yf zf]=rotation_z(xi,yi,zi,theta);
% function rf=rotation_y(ri);
%
% Estimates the coordinates of a given set of points when the y-axis is
% rotated by an angle theta clockwise by an angle theta
% (xi,yi,zi) are the coordinates of the points in the old coordinate system
% (xf,yf,zf) are the coordinates of the points in the new coordinate system
%
% rotation matrix
Rz=[cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1 ];

% rearrange points in a 3 x N matrix
a=size(xi);
ntot=prod(a);
xii=reshape(xi,1,ntot);
yii=reshape(yi,1,ntot);
zii=reshape(zi,1,ntot);
ri=[xii; yii; zii];
% rotate points

rf=Rz*ri;
% rearrange points in original dimensions
xff=rf(1,:);
yff=rf(2,:);
zff=rf(3,:);

xf=reshape(xff,a);
yf=reshape(yff,a);
zf=reshape(zff,a);

```



Εικόνα 14: Τα αρχικά σημεία και τα σημεία που προκύπτουν από την στροφή στους τρεις άξονες.

Πριν προχωρήσουμε περαιτέρω έχει ενδιαφέρον να δοκιμάσουμε την υλοποίηση των περιστροφών με το αρχείο *rotation_example* το οποίο περιστρέφει ένα σύνολο σημείων στο επίπεδο $z=0$ γύρω από τους τρεις άξονες x , y και z υπό γωνία $\pi/6$. Στην Εικόνα 14 δείχνουμε τις στροφές αυτές ενώ παρακάτω δίνουμε τον κώδικα του *rotation_example*.

rotation_example.m

```
% Rotation example

% setup a xy plane
close all
rmin_x=-1;
rmax_x=1;
rmin_y=-2;
rmax_y=2;
zp=0;
Nx=20;
Ny=20;
x=rmin_x:(rmax_x-rmin_x)/Nx:rmax_x;
y=rmin_y:(rmax_y-rmin_y)/Ny:rmax_y;
[xx,yy]=meshgrid(x,y);
zz=zp*ones(size(xx));

% draw original xy-plane
figure(1);
plot3(xx,yy,zz,'b. ');
set(gca,'FontSize',20);
xlabel('x'); ylabel('y'); zlabel('z');
theta=pi/3;
ylim([-2 2]);
xlim([-1 1]);

% rotate plane along x axis
[X,Y,Z]=rotation_x(xx,yy,zz,theta);

% draw rotated plane
figure(2);
plot3(X,Y,Z,'b. ');
set(gca,'FontSize',20);
xlabel('x'); ylabel('y'); zlabel('z');
ylim([-1 1]);
xlim([-1 1]);
% rotate plane along y axis
[X,Y,Z]=rotation_y(xx,yy,zz,theta);

% draw rotated plane
figure(3);
plot3(X,Y,Z,'b. ');
set(gca,'FontSize',20);
xlabel('x'); ylabel('y'); zlabel('z');
ylim([-2 2]);
xlim([-0.5 0.5]);
zlim([-0.5 0.5]);
% rotate plane along y axis
[X,Y,Z]=rotation_z(xx,yy,zz,theta);

% draw rotated plane
figure(4);
plot3(X,Y,Z,'b. ');
set(gca,'FontSize',20);
xlabel('x'); ylabel('y'); zlabel('z');
ylim([-1 1]);
xlim([-2 2]);
```

Βασισμένοι στην λογική που παρουσιάσαμε στην αρχή της παραγράφου υλοποιούμε την Gaussian δέσμη με γενικότερο προσανατολισμό με την συνάρτηση *gaussian_beam_arb*. Για να δοκιμάσουμε την υλοποίηση χρησιμοποιούμε το αρχείο *gaussian_beam_arb_example* με την οποία υπολογίζουμε την ένταση της Gaussian δέσμης που εκπέμπεται από ένα σημείο $(0,0,3)^t$ πάνω στον άξονα των z με διεύθυνση $\mathbf{n}_T = \frac{1}{3}(\sqrt{3}, \sqrt{3}, -\sqrt{3})^t$. Στην Εικόνα 15 δείχνουμε την γραφική παράσταση της έντασης του πεδίου όπως υπολογίζεται στην περίπτωση αυτή για το επίπεδο $z=0$. Όπως είναι αναμενόμενο, η ένταση είναι συγκεντρωμένη κοντά στο σημείο $(0,3,3)$ εξαιτίας του προσανατολισμού του διανύσματος \mathbf{n}_T .

gaussian_beam_arb.m

```
function [u,U]=gaussian_beam_arb(PT,xR,yR,zR,w0x,w0y,nT,r0,lambda);

% [u,U]=gaussian_beam_arb(PT,xR,yR,zR,w0x,w0y,r0,nT,lambda);

% Gaussian beam evolution estimation. Source is assumed to be located at
% r0 and the propagation is assumed along the nT vector
% PT is the transmitted power
% xR, yR, zR are the points in the 3D space where the Gaussian field is
% estimated.
% w0x is the initial 1/e^2 along the x-axis
% w0y is the initial 1/e^2 along the y-axis
% u is the Gaussian beam field
% U is the Gaussian beam intensity |u|^2

% First displace the axis so that the origin is at (0,0,0)
xR=xR-r0(1);
yR=yR-r0(2);
zR=zR-r0(3);

% Rotate the axis so that the vector (0,0,1) coincides with nT
nTc=cart_to_spher(nT);
thetaT=nTc(2);
phiT=nTc(3);

thetaT/pi
phiT/pi

% Rotate the axis so that the vector nT falls to z
[X,Y,Z]=rotation_z(xR,yR,zR,-phiT);
[X,Y,Z]=rotation_y(X,Y,Z,-thetaT);

[u,U]=gaussian_beam(PT,X,Y,Z,w0x,w0y,lambda);
flag=abs(Z>=0);
u=u.*flag;
U=U.*flag;
```

gaussian_beam_arb_example.m

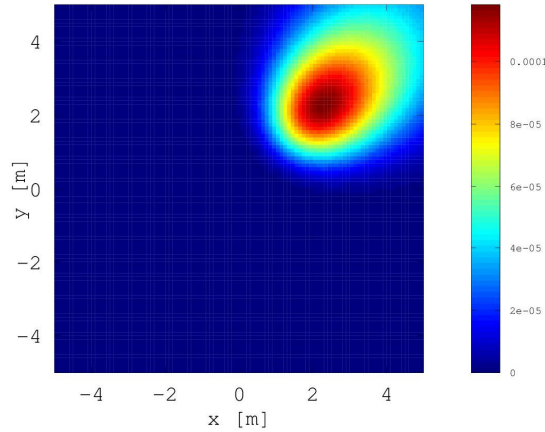
```
% Gaussian beam propagation example

w0x=1e-6;           % Beam width along x
w0y=1e-6;           % Beam width along y
L=3;                % propagation distance
lambda=1.55e-6;     % free space wavelength
PT=1e-3;            % transmission power
rmin=-5;
rmax=5;
Nr=100;
x=rmin:(rmax-rmin)/Nr:rmax;           % coordinates on the receiver plane
along the x-axis
y=rmin:(rmax-rmin)/Nr:rmax;           % coordinates on the receiver plane
along the y-axis
[xR,yR]=meshgrid(x,y);                % receiver plane x,y-coordinates
zR=zeros(size(xR));                    % receiver plane z-coordinates
nT=[+1 +1 -1];                          % propagation direction
r0=[0 0 L];                             % location of the source
```

```
[u,U]=gaussian_beam_arb(PT,xR,yR,zR,w0x,w0y,nT,r0,lambda); % estimation of the Gaussian
field

% Plot gaussian beam intensity
figure(1);
pcolor(xR,yR,U);
axis equal;
colorbar;
xlabel('x [m]');
set(gca,'FontSize',20);
ylabel('y [m]');
shading flat; axis tight;

% Plot gaussian beam angle
figure(2);
pcolor(xR,yR,angle(u));
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;
```



Εικόνα 15: Ένταση της Gaussian δέσμης όπως υπολογίζεται με το αρχείο *gaussian_beam_arb_example*.

5.4 Επίδραση των οπτικών στοιχείων του πομπού

Όπως είδαμε στην ενότητα 4.1 τα χαρακτηριστικά της οπτικής δέσμης καθορίζονται και από τα οπτικά στοιχεία του πομπού. Στην ενότητα 4.2 παρουσιάσαμε πως μπορούμε να υπολογίσουμε το εύρος μίας δέσμης μέσα από ένα γενικότερο οπτικό σύστημα. Ο παράγοντας q στην (7) παίζει ιδιαίτερο ρόλο στην εξέλιξη της ακτίνας και μπορούμε εύκολα να δούμε ότι καθορίζει και το ακριβές πεδίο της δέσμης.

Πράγματι όταν το αρχικό πεδίο είναι της μορφής $u_0 = \exp(-a_x x^2 - a_y y^2)$ και λαμβάνει χώρα στον άξονα των z τότε, σύμφωνα και με τα όσα είχαμε δει στο Π1.1 για να υπολογίσουμε το τελικό πεδίο u_1 υπολογίζουμε τον μετασχηματισμό Fourier του αρχικού πεδίου,

$$U_0(k_x, k_y) = \frac{\pi}{\sqrt{a_x a_y}} \exp\left(-\frac{k_x^2}{4a_x} - \frac{k_y^2}{4a_y}\right) \quad (25)$$

Ο μετασχηματισμός Fourier του τελικού πεδίου προκύπτει από τον πολλαπλασιασμό του U_0 με τον εκθετικό παράγοντα $\exp(-k_x^2/2/k - k_y^2/2/k)$ οπότε τελικά,

$$U_1(k_x, k_y) = \frac{\pi}{\sqrt{a_x a_y}} \exp\left(-\left[\frac{1}{4a_x} - j\frac{L}{2k}\right]k_x^2 - \left[\frac{1}{4a_y} - j\frac{L}{2k}\right]k_y^2\right) \quad (26)$$

ή αν θέσουμε

$$\frac{1}{4a'_x} = \frac{1}{4a_x} - j \frac{L}{2k} \quad (27)$$

$$\frac{1}{4a'_y} = \frac{1}{4a_y} - j \frac{L}{2k} \quad (28)$$

τότε θα έχουμε,

$$U_1(k_x, k_y) = \sqrt{\frac{a'_x a'_y}{a_x a_y}} \frac{\pi}{\sqrt{a'_x a'_y}} \exp\left(-\frac{k_x^2}{4a'_x} - \frac{k_y^2}{4a'_y}\right) \quad (29)$$

οπότε το πεδίο στο πεδίο του χώρου θα δίνεται από την σχέση:

$$u_1(x, y) = \sqrt{\frac{a'_x a'_y}{a_x a_y}} \exp(-a'_x x^2 - a'_y y^2) \quad (30)$$

Βλέπουμε επομένως ότι το πεδίο σε απόσταση L προκύπτει απευθείας από την (30) οι παράμετροι a'_x και a'_y προκύπτουν απευθείας από την (27)-(28). Εναλλακτικά μπορούμε να χρησιμοποιήσουμε τους παράγοντες q_x και q_y όπου

$$\frac{1}{q_x} = -j \frac{2a_x}{k} \quad (31)$$

$$\frac{1}{q_y} = -j \frac{2a_y}{k} \quad (32)$$

οπότε θα έχουμε:

$$q'_x = q_x + L \quad (33)$$

$$q'_y = q_y + L \quad (34)$$

ενώ καταλήγουμε στην σχέση:

$$u_1(x, y) = \sqrt{\frac{q_x q_y}{q'_x q'_y}} \exp\left(-\frac{jk}{2q'_x} x^2 - \frac{jk}{2q'_y} y^2\right) \quad (35)$$

Επίσης όταν η ακτίνα περνάει από έναν φακό αποκτά επιπλέον φάση $jk/2/f(x^2+y^2)$ οπότε στην περίπτωση αυτή θα έχουμε,

$$\frac{1}{q'_x} = \frac{1}{q_x} - \frac{1}{f} \quad (36)$$

$$\frac{1}{q'_y} = \frac{1}{q_y} - \frac{1}{f} \quad (37)$$

Στην περίπτωση αυτή η δέσμη δίνεται από την σχέση,

$$u_1(x, y) = \exp\left(-\frac{jk}{2q'_x} x^2 - \frac{jk}{2q'_y} y^2\right) \quad (38)$$

Επομένως συμπεραίνουμε ότι:

- όταν η δέσμη διαδίδεται στον ελεύθερο χώρο οι παράγοντες q σε κάθε διεύθυνση αλλάζουν σύμφωνα με τις (33)-(34) ενώ το πλάτος του πεδίου πολλαπλασιάζεται με τον παράγοντα $\sqrt{q_x q_y / q'_x / q'_y}$.
- όταν η δέσμη περνάει μέσα από έναν φακό τότε οι παράγοντες q σε κάθε διεύθυνση αλλάζουν σύμφωνα με τις (36)-(37) ωστόσο το πλάτος παραμένει αναλλοίωτο.

Η πληροφορία προκύπτει εν μέρει και από την θεωρία των ABCD πινάκων που είδαμε στην ενότητα 4.1. Η επιπλέον πληροφορία που έχουμε είναι ότι κάθε φορά που διαδίδεται η δέσμη στον ελεύθερο χώρο το πλάτος της πολλαπλασιάζεται με τον παράγοντα $\sqrt{q_x q_y / q'_x / q'_y}$ ο οποίος επηρεάζει τόσο την ένταση όσο και τη φάση του πεδίου. Η αλλαγή στην ένταση U οφείλεται στο ότι η δέσμη διευρύνεται και επομένως η ισχύς κατανέμεται σε μεγαλύτερη επιφάνεια οπότε η πυκνότητα ισχύος U γίνεται μικρότερη. Οι παραπάνω εξισώσεις μας παρέχουν έναν τρόπο υπολογισμού της διάδοσης της Gaussian δέσμης μέσα από ένα οπτικό σύστημα.

Ξεκινάμε όπως και πριν με την υλοποίηση της συνάρτησης `gaussian_beam_q` για τον υπολογισμό του πεδίου στην (35) και (38). Στη συνέχεια υλοποιούμε την σύγκριση των δύο μεθόδων στην περίπτωση της διάδοσης στον ελεύθερο χώρο, δηλαδή της `gaussian_beam` και της `gaussian_beam_q` με το αρχείο `gaussian_beam_q_example`.

gaussian_beam_q

```
function [u,U]=gaussian_beam_q(xR,yR,qx,qy,amp0,lambda)
% [u,U]=gaussian_beam_q(PT,xR,yR,zR,qx,qy,lambda)
%
% Gaussian beam estimation. Source is assumed to be oriented along the
% z-axis and the propagation is assumed along this axis
% xR, yR, are the points in the xy plane where the beam is to be estimated
% qx is the q-parameter along the x-axis
% qy is the q-parameter along the y-axis
% amp0 is the peak amplitude of the beam
% lambda is the wavelength
% u is the Gaussian beam field
% U is the Gaussian beam intensity |u|^2

k=2*pi/lambda; % free space propagation
u=amp0*exp(-j*k/2/qx*xR.^2-j*k/2/qy*yR.^2); % estimation of the beam
U=abs(u).^2; % Gaussian beam intensity
```

Gaussian_beam_q_example

```
% Example for gaussian_beam_q use
close all;
clear all;

w0x=2e-6; % Beam width along x
w0y=2e-6; % Beam width along y
L=5; % propagation distance
lambda=1.55e-6; % free space wavelength
PT=1e-3; % transmission power
rmin=-0.1;
rmax=0.1;
Nr=100;
x=rmin:(rmax-rmin)/Nr:rmax; % coordinates on the receiver plane
along the x-axis
y=2*rmin:(rmax-rmin)/Nr:2*rmax; % coordinates on the receiver plane
along the y-axis
[xR,yR]=meshgrid(x,y); % receiver plane x,y-coordinates
zR=L*ones(size(xR)); % receiver plane z-coordinates

% Calculate the Gaussian beam field the old way
[u1,U1]=gaussian_beam(PT,xR,yR,zR,w0x,w0y,lambda);

% Now calculate the Gaussian beam field using the gaussian_beam_q function

ampi=sqrt(2*PT/w0x/w0y/pi); % initial maximum amplitude

%Set up optical system for free space propagation
li=L; % single propagation distance
fi=[]; % no lens
[q,w1,R1,M1]=propagation_lens(li,fi,w0x,Inf,lambda); % q parameter of the beam
qx=q; % beam is symmetric
qy=q; % beam is symmetric
```

```

% Initial q parameters
q0inv=-j*lambda/pi/w0x^2;
q0x=1/q0inv;
q0y=1/q0inv;

% Beam amplitude for propagation in free space
amp0=ampi*sqrt(q0x*q0y/qx/qy);

% Estimate the Gaussian beam using gaussian_beam_q
[u2,U2]=gaussian_beam_q(xR,yR,qx,qy,amp0,lambda);

% Plot gaussian beam intensity
figure(1);
pcolor(xR,yR,U1);
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

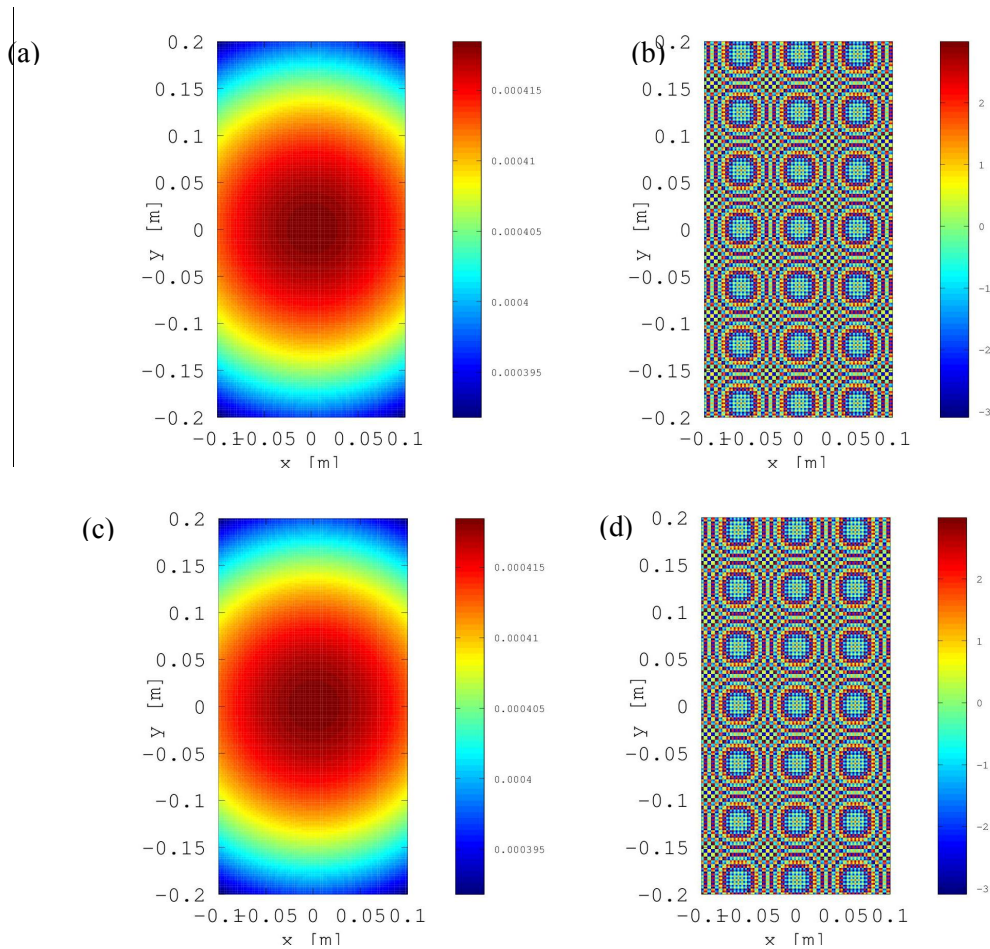
% Plot gaussian beam angle
figure(2);
pcolor(xR,yR,angle(u1));
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

% Plot gaussian beam intensity
figure(3);
pcolor(xR,yR,U2);
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

% Plot gaussian beam angle
figure(4);
pcolor(xR,yR,angle(u2));
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;

```

Στην Εικόνα 16 δείχνουμε την ένταση και την φάση της ακτίνας όπως αυτή υπολογίζεται πάνω σε μία επιφάνεια με διαστάσεις $0.2\text{m} \times 0.4\text{m}$ σε απόσταση $L=5\text{m}$ από την πηγή, θεωρώντας πως η επιφάνεια αυτή είναι προσανατολισμένη κάθετα στην διεύθυνση διάδοσης και πως το κέντρο της επιφάνειας βρίσκεται πάνω στην διεύθυνση μέγιστης μετάδοσης. Παρατηρούμε ότι οι συναρτήσεις *gaussian_beam_q* και *gaussian_beam* υπολογίζουν το ίδιο πεδίο και την ίδια φάση κάτι που ενισχύει την ορθότητα της υλοποίησης. Επίσης ιδιαίτερο ενδιαφέρον έχει η λεπτή υφή της φάσης κάτι το οποίο ωστόσο σχολιάσαμε και στο Π1.1 [2].



Εικόνα 16: Ένταση και φάση της Gaussian δέσμης όπως υπολογίζεται από την συνάρτηση *gaussian_beam* στις υποεικόνες (a) και (b) και *gaussian_beam_q* στις υποεικόνες (c) και (d).

Το επόμενο και τελικό βήμα είναι η υλοποίηση μίας συνάρτησης που θα υπολογίζει την ένταση και της φάσης μίας Gaussian δέσμης λαμβάνοντας υπόψη και το οπτικό σύστημα του πομπού και τον προσανατολισμό της δέσμης. Για το σκοπό αυτό υλοποιήσαμε αρχικά την συνάρτηση *qtowR* η οποία υπολογίζει τις παραμέτρους w και R από την παράμετρο q της δέσμης και την συνάρτηση *gaussian_beam_arbq* η οποία υπολογίζει την εν λόγω δέσμη. Το αρχείο *gaussian_beam_arbq_example* δείχνει ένα παράδειγμα χρήσης της εν λόγω συνάρτησης υπολογίζοντας εκ νέου την ένταση της Gaussian δέσμης που εκπέμπεται από ένα σημείο $(0,0,3)^T$ πάνω στον άξονα των z με διεύθυνση $\mathbf{n}_T = \frac{1}{3}(\sqrt{3}, \sqrt{3}, -\sqrt{3})^T$. Στην δείχνουμε την γραφική παράσταση της έντασης του πεδίου όπως υπολογίζεται στην περίπτωση αυτή για το επίπεδο $z=0$.

qtowR.m

```
function [w,R]=qtowR(q,lambda);
% function [w,R]=qtowR(q);
%
% estimates the width w and the curvature radius of a Gaussian beam from
% its q parameter

qr=real(1./q);
qi=imag(1./q);

%Estimate beam width and beam radius
R=1./qr;
w=sqrt(-lambda/pi./qi);
```

gaussian_beam_arbq.m

```
function [u,U]=gaussian_beam_arbq(PT,xR,yR,zR,w0x,w0y,nT,r0,lambda,li,fi);
```

```

% [u,U]=gaussian_beam_arbq(PT,xR,yR,zR,w0x,w0y,r0,nT,lambda,li,fi);

% Gaussian beam evolution estimation. Source is assumed to be located at
% r0 and the propagation is assumed along the nT vector. For this
% implementation the source position is taken as the output point of the
% optical system
%
% PT is the transmitted power
% xR, yR, zR are the points in the 3D space where the Gaussian field is
% estimated.
% w0x is the initial 1/e^2 along the x-axis
% w0y is the initial 1/e^2 along the y-axis
% u is the Gaussian beam field
% U is the Gaussian beam intensity |u|^2
% li and fi describe the optical transmitter system

% First displace the axis so that the origin is at (0,0,0)
xR=xR-r0(1);
yR=yR-r0(2);
zR=zR-r0(3);

% Rotate the axis so that the vector (0,0,1) coincides with nT
nTc=cart_to_spher(nT);
thetaT=nTc(2);
phiT=nTc(3);

% Rotate the axis so that the vector nT falls to z
[X,Y,Z]=rotation_z(xR,yR,zR,-phiT);
[X,Y,Z]=rotation_y(X,Y,Z,-thetaT);

% Estimate the q parameters of the final beam for every point on the Z-axis
sz=size(Z);
Z=reshape(Z,numel(Z),1);
qx=zeros(size(Z));
qy=zeros(size(Z));

for m=1:length(Z)
    [qx(m),~,~,~]=propagation_lens([li Z(m)],fi,w0x,Inf,lambda); % qx parameter of the
beam
    [qy(m),~,~,~]=propagation_lens([li Z(m)],fi,w0y,Inf,lambda); % qy parameter of the
beam
end

Z=reshape(Z,sz);
qx=reshape(qx,sz);
qy=reshape(qy,sz);

% Estimate the final beamwidths along x and along y
[wX,R1]=qtowR(qx,lambda);
[wY,R2]=qtowR(qy,lambda);

% Estimate final peak amplitude
amp0=sqrt(2*PT./wX./wY/pi);

% Estimate gaussian beam
[u,U]=gaussian_beam_q(X,Y,qx,qy,amp0,lambda)
flag=abs(Z>=0);
u=u.*flag;
U=U.*flag;

```

gaussian_beam_arbq_example.m

```

% Gaussian beam propagation example

w0x=1e-6;           % Beam width along x
w0y=1e-6;           % Beam width along y
L=3;                % propagation distance
lambda=1.55e-6;     % free space wavelength
PT=1e-3;            % transmission power

```

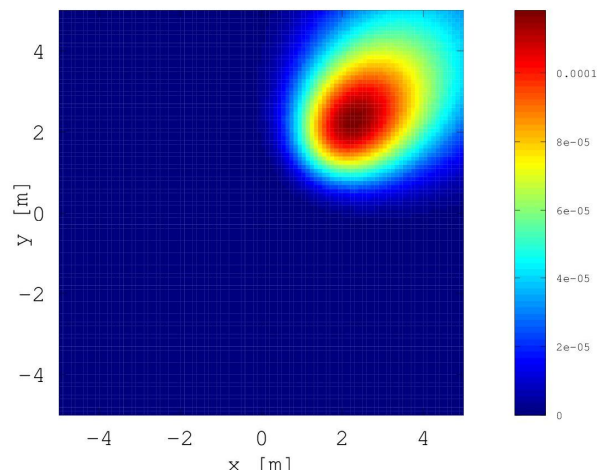
```

rmin=-5;
rmax=5;
Nr=100;
x=rmin:(rmax-rmin)/Nr:rmax;           % coordinates on the receiver plane
along the x-axis
y=rmin:(rmax-rmin)/Nr:rmax;           % coordinates on the receiver plane
along the y-axis
[xR,yR]=meshgrid(x,y);                % receiver plane x,y-coordinates
zR=zeros(size(xR));                   % receiver plane z-coordinates
nT=[+1 +1 -1];                         % propagation direction
r0=[0 0 L];                            % location of the source
li=[];
fi=[];
[u,U]=gaussian_beam_arbq(PT,xR,yR,zR,w0x,w0y,nT,r0,lambda,li,fi);

% Plot gaussian beam intensity
figure(1);
pcolor(xR,yR,U);
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('\itx\rm [m]');
ylabel('\ity\rm [m]');
shading flat; axis tight;

% Plot gaussian beam angle
figure(2);
pcolor(xR,yR,angle(u));
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('\itx\rm [m]');
ylabel('\ity\rm [m]');
shading flat; axis tight;

```



Εικόνα 17: Ένταση της Gaussian δέσμης όπως υπολογίζεται με το αρχείο *gaussian_beam_arbq_example*.

6. Ισολογισμός Ισχύος

6.1 Θεωρητικό υπόβαθρο

Έχοντας υλοποιήσει τον υπολογισμό του πεδίου και της έντασης του, είμαστε έτοιμοι να προχωρήσουμε στην υλοποίηση του ισολογισμού ισχύος της ζεύξης. Όπως είδαμε και στο Π1.1, το δiάνυσμα Poynting συνδέεται με την ένταση του κύματος σύμφωνα με την σχέση,

$$\mathbf{W} = |u|^2 \hat{\mathbf{e}} \quad (39)$$

όπου $\hat{\mathbf{e}}$ είναι ένα μοναδιαίο δiάνυσμα στην διεύθυνση της ευθείας που συνδέει τον πομπό και το δέκτη με κατεύθυνση από τον πομπό προς το δέκτη. Για να υπολογίσουμε την οπτική ισχύ που διέρχεται από μία επιφάνεια με εμβαδόν A_R θα πρέπει να λάβουμε υπόψη τον προσανατολισμό της επιφάνειας όπως αυτή εκφράζεται από το κάθετο σε αυτήν δiάνυσμα \mathbf{n}_R . Η συνολική ισχύς P_R δίνεται από το ολοκλήρωμα του $\mathbf{W} \cdot \mathbf{n}_R$ πάνω στην επιφάνεια του δέκτη που αν θεωρήσουμε ότι βρίσκεται αρκετά μακριά από τον πομπό θα είναι σχεδόν σταθερό πάνω στην επιφάνεια, με συνέπεια

$$P_R \cong \mathbf{W} \cdot \mathbf{n}_R A_R = |u|^2 A_R \cos \theta_R \quad (40)$$

όπου θ_R είναι η γωνία μεταξύ των διανυσμάτων \mathbf{n}_R και $\hat{\mathbf{e}}$ οπότε θα έχουμε:

$$\cos \theta_R = \frac{(\mathbf{r}_R - \mathbf{r}_T) \cdot \mathbf{n}_R}{|\mathbf{r}_R - \mathbf{r}_T|} \quad (41)$$

Στην περίπτωση όπου έχουμε περισσότερους πομπούς, απλά προσθέτουμε την ισχύ που φτάνει από κάθε έναν από αυτούς. Θα πρέπει να σημειωθεί ότι αυτή η παραδοχή σημαίνει ότι δεν υπάρχει χρονική διαφορά φάσης ανάμεσα από τις δέσμες των πολλαπλών πομπών ή ότι αυτή έχει εξαλειφθεί με αποτέλεσμα να έχουμε θετική συμβολή των διαφόρων κυμάτων.

6.2 Υλοποίηση

Για τον υπολογισμό του ισολογισμού ισχύος χρησιμοποιούμε την συνάρτηση `power_budget` η οποία υπολογίζει την ισχύ του δέκτη σε διάφορες θέσεις μέσα στον χώρο. Το αρχείο `power_budget_example` δείχνει ένα παράδειγμα τρόπου χρήσης της συνάρτησης `power_budget`.

power_budget.m

```
function [P,U]=power_budget(rT,nT,PT,fiT,liT,lambda,w,rR,nR,AR);
% P=power_budget(transmitter,receiver);
%
% Estimates the receiver power obtained at various receiver points
% rT are the transmitter positions. In the case of multiple transmitter we
% simply add the received powers
% nT are the transmitter orientations.
% PT are the transmitted powers for each transmitter
% fiT are the focal lengths of the transmitter lenses
% liT are the interlens spacings
% lambda is the wavelength
% w is the original beam width (assumed symmetrical)
% rR are the receiver positions
% nR are the receiver orientations
% AR are the receiver effective areas

[Nt,Yt]=size(rT);
[Nr,Yr]=size(rR);
xR=rR(:,1);
yR=rR(:,2);
zR=rR(:,3);
P=zeros(size(xR));

% normalize all transmitter orientation vectors
for nt=1:Nt
```

```

    nT(nt,:) = nT(nt,:) / norm(nT(nt,:));
end

% normalize all receiver orientation vectors
for nr=1:Nr
    nR(nr,:) = nR(nr,:) / norm(nR(nr,:));
end

% For every transmitter, estimate field intensities at each receiver point
for nt=1:Nt
    PT1=PT(nt); % Transmitter power
    nT1=nT(nt,:); % Transmitter orientation
    w1=w(nt,:); % Beamwidth at transmitter
    fi1=fiT(nt); % transmitter lens foci
    li1=liT(nt); % transmitter lens path length
    r1=rT(nt,:); % transmitter position
    [u1,U]=gaussian_beam_arbq(PT1,xR,yR,zR,w1,w1,nT1,r1,lambda,li1,fi1); % Estimate field
intensity
    a=rT(nt*ones(Nr,1),:)-rR; % Estimate propagation vector
for each receiver
    nrm=sqrt(a(:,1).^2+a(:,2).^2+a(:,3).^2); % normalization coefficient
    a(:,1)=a(:,1)./nrm; % normalize propagation vectors
    a(:,2)=a(:,2)./nrm;
    a(:,3)=a(:,3)./nrm;
    dotp=a(:,1).*nR(:,1)+a(:,2).*nR(:,2)+a(:,3).*nR(:,3); % dot product of a's and nR's
    P=P+U.*AR.*dotp;
end

```

power_budget_example.m

```

% An example to demonstrate the use of power_budget
%
close all;
clear all;

% Transmitter element
rT=[+2.5 +2.5 3]; % transmitter position
nT=[0 0 -1]; % transmitter orientation
PT=1e-3; % transmit power
fiT=cell(1,1); % transmitter lenses
liT=cell(1,1); % spacing between transmitter lenses
lambda=1.55e-6; % wavelength of operation
w=1e-6; % initial beamwidth

% Setup receiver elements
rmin=-5.0;
rmax=5.0;
Nr=100;
x=rmin:(rmax-rmin)/Nr:rmax; % coordinates on the receiver plane
along the x-axis
y=rmin:(rmax-rmin)/Nr:rmax; % coordinates on the receiver plane
along the y-axis
[xR,yR]=meshgrid(x,y); % receiver plane x,y-coordinates
zR=zeros(size(xR)); % receiver plane z-coordinates
szx=size(xR);
xR=reshape(xR,numel(xR),1);
yR=reshape(yR,numel(yR),1);
zR=reshape(zR,numel(zR),1);
rR=[xR yR zR];
nR1=zeros(numel(xR),1);
nR2=zeros(numel(xR),1);
nR3=ones(numel(xR),1);
nR=[nR1 nR2 nR3];
AR=1e-2;
[P,U]=power_budget(rT,nT,PT,fiT,liT,lambda,w,rR,nR,AR);

% reshape P

```

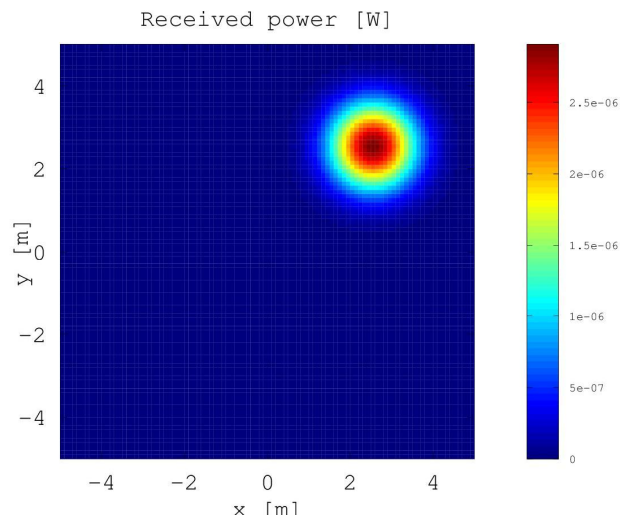


```

P=reshape(P,szx);
U=reshape(U,szx);
xR=reshape(xR,szx);
yR=reshape(yR,szx);

figure(1);
pcolor(xR,yR,P);
axis equal;
colorbar;
set(gca,'FontSize',20);
xlabel('x [m]');
ylabel('y [m]');
shading flat; axis tight;
title('Received power [W]');

```



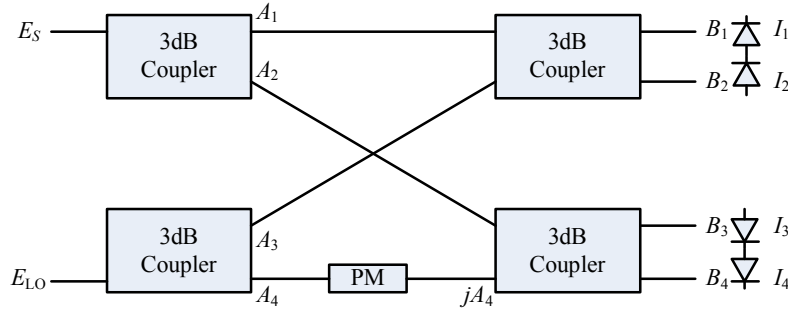
Εικόνα 18: Ισχύς λήψης όπως αυτή υπολογίζεται από το *power_budget_example*.

Στην Εικόνα 18 υπολογίζουμε την λαμβανόμενη ισχύ από το *power_budget_example* που αντιστοιχεί σε έναν πομπό ο οποίος είναι τοποθετημένος στις συντεταγμένες $(+2.5\text{m}, +2.5\text{m}, 3\text{m})$ στο επίπεδο του δέκτη $z=0$ για ισχύ εκπομπής $P_T=1\text{mW}$. Ο δέκτης είναι προσανατολισμένος προς τα επάνω, $\mathbf{n}_R=(0\ 0\ 1)^t$ και ο πομπός είναι προσανατολισμένος προς τα κάτω, $\mathbf{n}_T=(0\ 0\ -1)^t$.

7. Οπτικός Δέκτης

7.1 Θεωρητικό Υπόβαθρο

Ο σύμφωνος οπτικός δέκτης την καρδιά του σύμφωνου συστήματος. Η σωστή μοντελοποίηση του περιλαμβάνει και την μοντελοποίηση των διαφόρων θορύβων πλάτους και φάσης και εξαρτάται σε μεγάλο βαθμό από το είδος του δέκτη που χρησιμοποιείται. Στην παρούσα ανάλυση θα χρησιμοποιήσουμε το υπόδειγμα του δέκτη που παρουσιάσαμε στο Π1.1 [2] και παρουσιάζεται στο παρακάτω σχήμα:



Εικόνα 19: Ένας σύμφωνος δέκτης που βασίζεται σε ένα οπτικό υβρίδιο.

Το πεδίο του σήματος E_S και το πεδίο του τοπικού ταλαντωτή E_{LO} στην είσοδο του δέκτη μπορεί να γραφτεί ως εξής:

$$E_R = \left\{ \frac{\sqrt{P_R}}{2} \sin\left(\pi \frac{v_i}{V_\pi}\right) + j \frac{\sqrt{P_R}}{2} \sin\left(\pi \frac{v_q}{V_\pi}\right) \right\} e^{j\Delta\phi} + E_n = (E_I + jE_Q) e^{j\Delta\phi} + E_n = E_S e^{j\Delta\phi} + E_n \quad (42)$$

όπου έχουμε ορίσει τα πεδία:

$$E_I = \frac{\sqrt{P_R}}{2} \sin\left(\pi \frac{v_i}{V_\pi}\right) = \frac{\sqrt{P_R}}{2} e_I \quad (43)$$

$$E_Q = \frac{\sqrt{P_R}}{2} \sin\left(\pi \frac{v_q}{V_\pi}\right) = \frac{\sqrt{P_R}}{2} e_Q \quad (44)$$

$$E_S = E_I + jE_Q \quad (45)$$

όπου έχουμε χρησιμοποιήσει την (5) έχοντας αντικαταστήσει την ισχύ εκπομπής P_T με την ισχύ λήψης P_R , έχοντας ολισθήσει την φάση του σήματος κατά $\Delta\phi$ εξαιτίας του θορύβου φάσης¹ ή τυχαίων μικροδονήσεων και έχουμε προσθέσει ένα πεδίο που οφείλεται στον οπτικό θόρυβο που μπορεί να προέρχεται από το φωτισμό του περιβάλλοντος [5] ή σε φαινόμενα πολλαπλής διαδρομής. Τα ηλεκτρικά ρεύματα που παράγονται στον κάθε ένα φωτοδέκτη από τις σχέσεις:

$$i_1 = \frac{R}{4} \left(|E_R|^2 + |E_{LO}|^2 - 2 \operatorname{Re}\{E_{LO}^* E_R\} \right) + n_1 \quad (46)$$

$$i_2 = \frac{R}{4} \left(|E_R|^2 + |E_{LO}|^2 + 2 \operatorname{Re}\{E_{LO}^* E_R\} \right) + n_2 \quad (47)$$

$$i_3 = \frac{R}{4} \left(|E_R|^2 + |E_{LO}|^2 + 2 \operatorname{Im}\{E_{LO}^* E_R\} \right) + n_3 \quad (48)$$

¹ Στην περίπτωση του θορύβου φάσης θα πρέπει να λάβουμε υπόψη τόσο τον θόρυβο φάσης του σήματος όσο και του τοπικού ταλαντωτή

$$i_4 = \frac{R}{4} \left(|E_R|^2 + |E_{LO}|^2 - 2 \operatorname{Im} \{ E_{LO}^* E_R \} \right) + n_4 \quad (49)$$

Στην συνέχεια θα πρέπει να υπολογίσουμε τις προσθετικές συνιστώσες θορύβου στις (46)-(49) οι οποίες οφείλονται στο θόρυβο shot και στον θερμικό θόρυβο. Χρησιμοποιώντας τις εξισώσεις που είδαμε στο Π1.1 θα έχουμε τις εξής φασματικές πυκνότητες θορύβου για τον θερμικό και τον θόρυβο shot σε κάθε φωτοδίοδο:

$$S_T(f) = \frac{4k_B T}{R_F} + \frac{16\pi^2 k_B T}{g_m} \left(\Gamma + \frac{1}{g_m R_D} \right) C_T^2 f^2 + \frac{4\pi^2 K I_D C_T^2 f}{g_m^2} \quad (50)$$

$$S_{SH}(f) \cong \frac{qR}{2} (P_{LO} + P_n) \quad (51)$$

Στην (51) έχουμε χρησιμοποιήσει το γεγονός ότι κατά κανόνα η ισχύς του τοπικού ταλαντωτή P_{LO} είναι πολύ μεγαλύτερη από την ισχύ του σήματος P_R και επομένως η συνεισφορά της μπορεί να αγνοηθεί στην (51). Παράλληλα σε κάθε φωτοδίοδο προσπίπτει $\frac{1}{4}$ της οπτικής ισχύος και επομένως αντί του συνηθισμένου όρου $2qR$ χρησιμοποιούμε τον συντελεστή $qR/2$ για τον υπολογισμό της φασματικής πυκνότητας ισχύος. Η σημασία των παραμέτρων στην (50) έχουν εξηγηθεί στο Π1.1 [2]. Η ισχύς των ρευμάτων του θορύβου δίνεται από την

$$\sigma_n^2 = \langle n_i^2 \rangle = \int_0^B [S_T(f) + S_{SH}(f)] df \quad (52)$$

Στην συνέχεια υπολογίζουμε τα διαφορικά ρεύματα,

$$i_I = i_1 - i_2 = R \operatorname{Re} \{ E_{LO}^* E_R \} + n_I \quad (53)$$

$$i_Q = i_3 - i_4 = R \operatorname{Im} \{ E_{LO}^* E_R \} + n_Q \quad (54)$$

όπου

$$n_I = n_1 - n_2 \quad (55)$$

$$n_Q = n_3 - n_4 \quad (56)$$

Η ισχύς των συνιστωσών των θορύβων είναι

$$\langle n_I^2 \rangle = \langle n_Q^2 \rangle = 2\sigma_n^2 \quad (57)$$

Στη συνέχεια αναπτύσσουμε τις (53)-(54) ώστε να καταλήξουμε στις σχέσεις που συνδέουν τα ρεύματα i_I και i_Q

$$i_I = R E_{LO} \{ E_I \cos(\Delta\phi) - E_Q \sin(\Delta\phi) + E_{nr} \} + n_I \quad (58)$$

$$i_Q = R E_{LO} \{ E_I \sin(\Delta\phi) + E_Q \cos(\Delta\phi) + E_{ni} \} + n_Q \quad (59)$$

όπου έχουμε θέσει $E_{nr} = \operatorname{Re} \{ E_n \}$ και $E_{ni} = \operatorname{Im} \{ E_n \}$ ενώ έχουμε θεωρήσει ότι το πεδίο του τοπικού ταλαντωτή είναι πραγματικό. Αντικαθιστώντας τις εκφράσεις των πεδίων στην (43)-(44) βρίσκουμε,

$$i_I = \frac{R\sqrt{P_{LO}P_R}}{2} \{ e_I \cos(\Delta\phi) - e_Q \sin(\Delta\phi) \} + R\sqrt{P_{LO}} E_{nr} + n_I \quad (60)$$

$$i_Q = \frac{R\sqrt{P_{LO}P_R}}{2} \{ e_I \sin(\Delta\phi) + e_Q \cos(\Delta\phi) \} + R\sqrt{P_{LO}} E_{ni} + n_Q \quad (61)$$

Εδώ θα πρέπει να σημειώσουμε ότι υπάρχει στις (60)-(61) προστίθενται επιπλέον συνιστώσες θορύβου

$$\Delta n_I = R\sqrt{P_{LO}} E_{nr} \quad (62)$$

$$\Delta n_Q = R\sqrt{P_{LO}} E_{ni} \quad (63)$$

που οφείλονται στην μίξη του θορύβου περιβάλλοντος και του τοπικού ταλαντωτή. Ωστόσο θα πρέπει να σημειωθεί ότι η περιβάλλουσα ακτινοβολία είναι *ασύμφορη* και επομένως δεν μπορεί να αναμειχθεί με τον τοπικό ταλαντωτή. Επομένως έχουμε,

$$i_I = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_I \cos(\Delta\phi) - e_Q \sin(\Delta\phi)\} + n_I \quad (64)$$

$$i_Q = \frac{R\sqrt{P_{LO}P_R}}{2} \{e_I \sin(\Delta\phi) + e_Q \cos(\Delta\phi)\} + n_Q \quad (65)$$

7.2 Υπολογισμός της ισχύος του προσθετικού θορύβου

Ο υπολογισμός της ισχύος του θορύβου βασίζεται στην αριθμητική ολοκλήρωση της φασματικής πυκνότητας ισχύος στο εύρος ζώνης του σήματος ή του δέκτη B . Αυτό το έχουμε υλοποιήσει με την συνάρτηση *noise_variance* την οποία παραθέτουμε και παρακάτω. Ένα παράδειγμα χρήσης της συνάρτησης *noise_variance* είναι το αρχείο *noise_example* το οποίο επίσης παραθέτουμε. Στην Εικόνα 20 έχουμε παρουσιάσει τον υπολογισμό της ισχύος του θερμικού θορύβου και του θορύβου βολής καθώς και την συνολική ισχύ του θορύβου.

noise_variance.m

```
function [sv,svt,svs,f3dB]=noise_variance(receiver,PR,pn);
%
% sv=noise_variance(receiver,lambda,pn)
%
% Calculates the noise variance of the additive noise of a coherent
% receiver

% Constants
kB=1.38e-23; % Boltzmann constant (m^2*Kg/s^2/K)
T=300; % Temperature (K)
q=1.6e-23; % Electron charge density
c=3e8; % speed of light

% Receiver parameters (assuming a MOS TIA)
RF=receiver.RF; % Feedback resistance
Cd=receiver.Cd; % photodiode capacitance
Cds=receiver.Cds; % MOS drain/source capacitance
Cgs=receiver.Cgs; % MOS gate/source capacitance
gm=receiver.gm; % MOS transconductance
ID=receiver.ID; % MOS DC drain current
K=receiver.K; % MOS K parameter
R=receiver.R; % detector responsivity
RD=receiver.RD; % drain resistance
gamma=receiver.gamma; % MOS channel noise factor
PLO=receiver.PLO; % Receiver LO power
A2=receiver.A2; % amplifier gain
B=receiver.B; % electrical receiver bandwidth
Dl=receiver.Dl; % receiver filter optical bandwidth
AR=receiver.AR; % receiver area

% Estimate 3dB bandwidth
CT=Cd+Cds+Cgs;
f3dB=(gm*RD*A2+1)/(2*pi*RF*CT);

% Estimate power spectral density of thermal noise
f=0:B/1000:B;
Sthermal1 = 4*kB*T/RF*ones(size(f));
Sthermal2 = 16*pi^2*kB*T/gm * (gamma+1/gm/RD)*CT^2*f.^2;
Sthermal3 = 4*pi^2*K*ID*CT^2*f/gm^2;
ST=Sthermal1+Sthermal2+Sthermal3;

% Estimate power spectral density of shot noise
Pn=pn*Dl*AR;
SH=q*R*(PLO+PR+Pn)*ones(size(ST));
```

```

% Numerically estimate the noise variance using the trapezium rule
svt=trapz(f,ST);
svs=trapz(f,SH);
sv=svt+svs;

```

noise_example.m

```

% Example demonstrating the use of noise_variance

% Receiver parameters
receiver.RF=10e3;           % Feedback resistance
receiver.Cd=20e-12;        % photodiode capacitance
receiver.Cds=0.5e-12;     % MOS drain/source capacitance
receiver.Cgs=0.5e-12;     % MOS gate/source capacitance
receiver.gm=40e-3;        % MOS transconductance
receiver.ID=20e-3;        % MOS DC drain current
receiver.K=294e-15;       % MOS K parameter
receiver.R=0.53;          % detector responsivity
receiver.RD=1000;         % drain resistance
receiver.gamma=1;         % MOS channel noise factor
receiver.B=3e8;           % electrical receiver bandwidth
receiver.PLO=1e-3;        % Receiver LO power
receiver.A2=100;          % amplifier gain
receiver.Dl=40e-9;       % electrical receiver bandwidth
receiver.AR=1e-2;         % receiver area

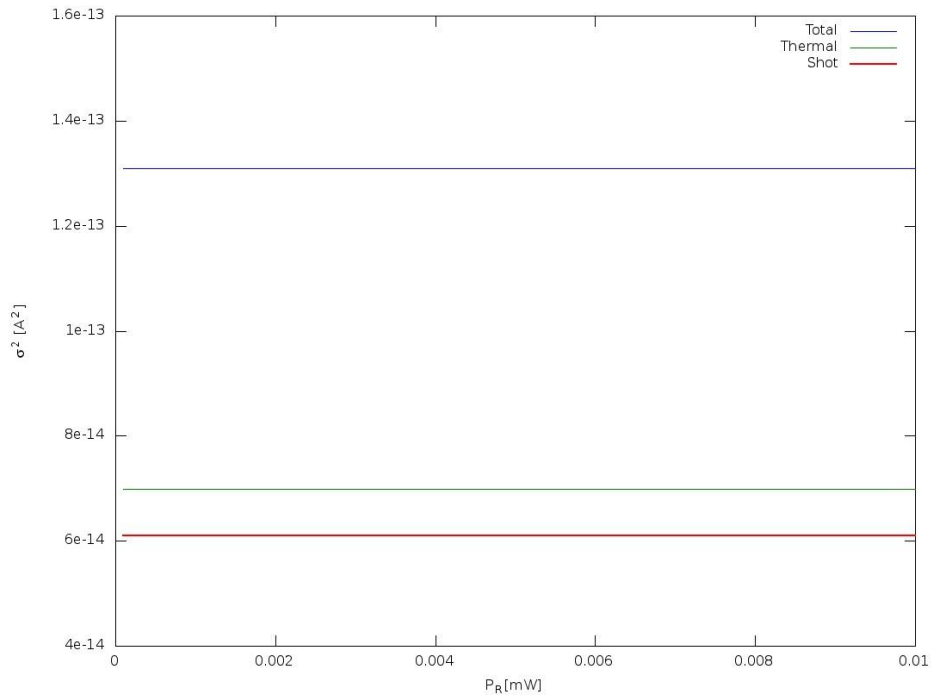
% Siganl and ambient noise parameters
lambda=1.55e-6;           % Wavelength
pn=6e10;                  % ambient light irradiance
PR=1e-7:1e-7:1e-5;       % Received power

sv=zeros(1,length(PR));
svt=zeros(1,length(PR));
svs=zeros(1,length(PR));

for m=1:length(PR)
    [sv(m),svt(m),svs(m)]=noise_variance(receiver,PR(m),pn);
end

figure(1);
plot(PR/1e-3,sv,PR/1e-3,svt,PR/1e-3,svs,'LineWidth',2);
set(gca,'FontSize',20);
xlabel('P_R[mW]');
ylabel('\sigma^2 [A^2]');
legend('Total','Thermal','Shot');

```



Εικόνα 20: Υπολογισμός της ισχύος των θορύβων από την *noise_example*

7.3 Υπολογισμός του προσθετικού θορύβου

Στη συνέχεια θα πρέπει να υλοποιήσουμε την δημιουργία της κυματομορφής του προσθετικού θορύβου. Εδώ θα πρέπει να διαχωρίσουμε δύο περιπτώσεις: σε μία πρώτη προσέγγιση τα δείγματα του θορύβου δεν παρουσιάζουν συσχέτιση οπότε ο θόρυβος είναι Gaussian λευκός, προσθετικός (Additive White Gaussian Noise – AWGN). Η συγκεκριμένη περίπτωση υλοποιείται εύκολα με την συνάρτηση *myawgn* της οποίας ο κώδικας φαίνεται παρακάτω:

myawgn.m

```
function n=myawgn(t,sv);
% function n=awgn(t,sv);
%
% Generates the samples of an AWGN noise
% sv is the power of the noise
% t is the time axis

n=sqrt(sv)*randn(size(t));
```

Ιδιαίτερο ενδιαφέρον παρουσιάζει και η περίπτωση όπου τα δείγματα του θορύβου παρουσιάζουν μία αυτοσυσχέτιση και επομένως δεν έχουμε να κάνουμε με λευκό αλλά έγχρωμο θόρυβο. Στην περίπτωση αυτή χρησιμοποιούμε μία εναλλακτική και λίγο πιο σύνθετη υλοποίηση όπου γεννούμε δείγματα λευκού θορύβου και στην συνέχεια τα φιλτράρουμε τα δείγματα αυτά με μία συνάρτηση $H(f) = \sqrt{S(f)}$ όπου $S(f)$ είναι η φασματική πυκνότητα του θορύβου που θέλουμε να πετύχουμε. Για τον σκοπό αυτό χρησιμοποιούμε την συνάρτηση *colored_noise* που δείχνουμε παρακάτω

colored_noise.m

```
function [n,f]=colored_noise(t,fi,Si);
% [n,f]=colored_noise(t,fi,Si);
%
% generation of colored noise samples
% Si is the power spectral density
% fi is the frequency axis
% t is the time axis
```

```

Dt=t(2)-t(1); % estimate time increment
Nt=length(t); % number of samples to be generated
Df=1/Nt/Dt; % frequency spacing for FFT
f=(-Nt/2+1:Nt/2-1)*Df; % frequency axis for FFT
PSD=interp1(fi,Si,f); % interpolated PSD at the FFT frequency axis
i=find(~isfinite(PSD)); % search for NaNs
PSD(i)=0; % remove NaNs from interpolated PSD
H=sqrt(abs(PSD)); % filter response
s=randn(1,Nt); % generate white samples
S=fftshift(fft(fftshift(s))); % estimate spectrum
S=S.*H; % apply coloring
s2=fftshift(iff(fftshift(S))); % go back to the time domain
n=s2/sqrt(Dt); % normalize samples

```

Το αρχείο *noise_correlation* είναι ένα παράδειγμα χρήσης της συνάρτησης *colored_noise* όπου δοκιμάζουμε κατά πόσο η συνάρτηση μπορεί να παράγει δείγματα με Gaussian φασματική πυκνότητα.

noise_correlation.m

```

% test noise correlation of colored_noise
Nt=65536; % number of samples
Nf=64; % number of frequency points for PSD
Dt=0.001; % time step
iterations=1e4; % number of sample generation iterations
t=(-Nt/2+1:(Nt/2-1))*Dt; % time axis
B=1; % 1/e bandwidth of the PSD
Df=10*B/Nf; % PSD frequency axis spacing
fi=(-Nf/2+1:(Nf/2-1))*Df; % frequency axis
Si=exp(-fi.^2/2/B^2); % Gaussian PSD
corr=zeros(1,length(t)); % to hold the correlation of the samples

% start iterations and calculate correlation numerically
for m=1:iterations
    [x,f]=colored_noise(t,fi,Si);
    corr=corr+1/iterations*x(Nt/2)*x;
end

figure(1);
plot(t,corr,'LineWidth',2);
set(gca,'FontSize',20);
xlabel('t[s]');
ylabel('Rnn(t)');
xlim([-4/B 4/B]);

figure(2);
Scorr=Dt*fftshift(fft(fftshift(corr)));
plot(f,abs(Scorr),fi,abs(Si),'r','LineWidth',2);
set(gca,'FontSize',20);
xlabel('f[Hz]');
ylabel('S(f)');
xlim([-4*B 4*B]);
legend('numerical','specified');

figure(3);
plot(t,real(x(1,:)),t,imag(x(1:)), 'LineWidth',2);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('n(t)');
legend('real part','imaginary part');

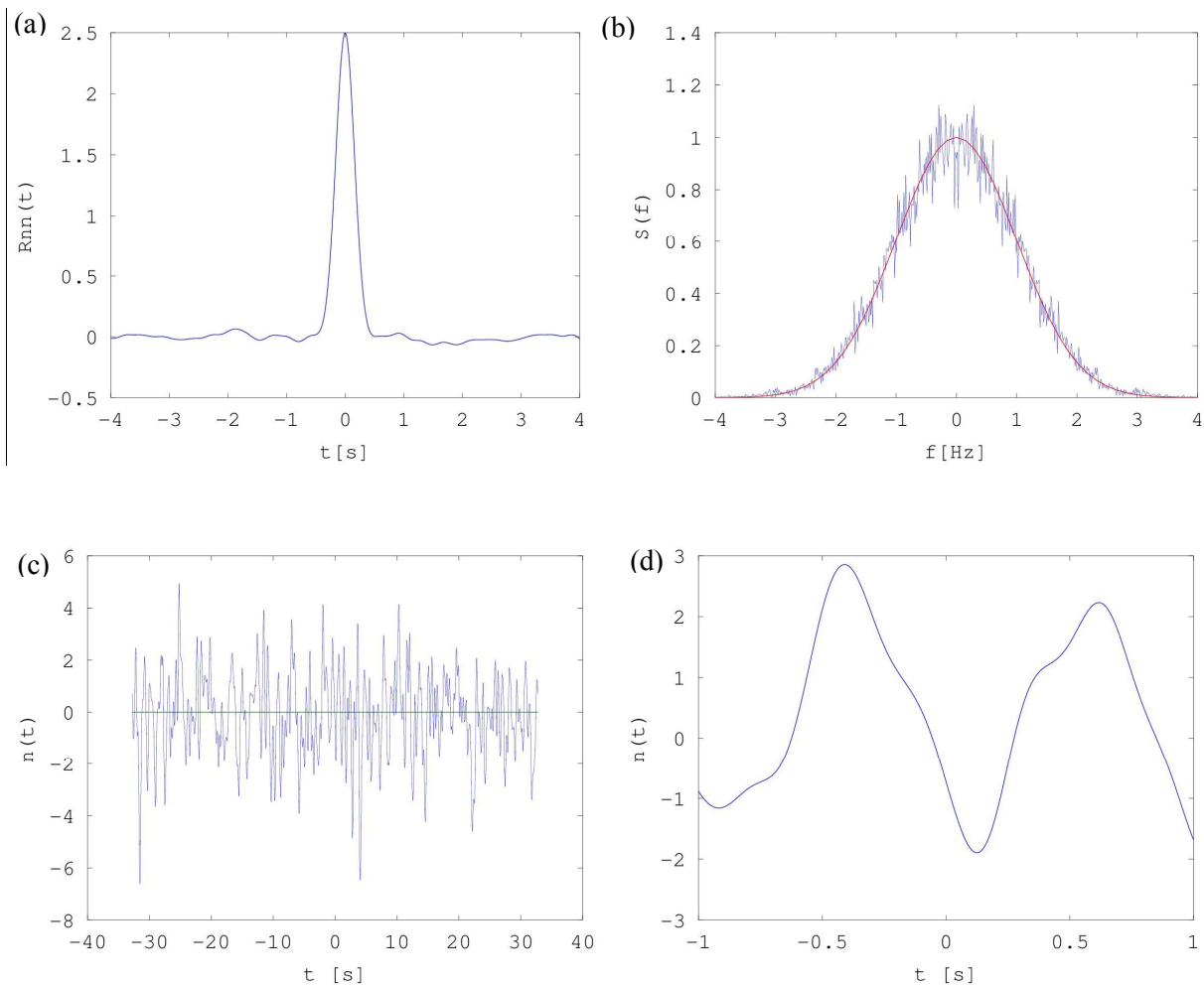
figure(4);
plot(t,real(x(1:)), 'LineWidth',2);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('n(t)');
xlim([-1 1]);

```

```
ylim([-3 3]);
```

Στις παρακάτω εικόνες παραθέτουμε τα στατιστικά χαρακτηριστικά των δειγμάτων της *colored_noise* όπως αυτά υπολογίζονται με το αρχείο *noise_correlation*. Παρατηρούμε ότι η αριθμητικά υπολογισμένη συνάρτηση αυτοσυσχέτισης είναι Gaussian ενώ το ίδιο ισχύει και με την φασματική πυκνότητα ισχύος η οποία συμφωνεί με την αρχική οριζόμενη Gaussian συνάρτηση (κόκκινη καμπύλη). Επίσης παρατηρούμε ότι τα δείγματα δεν έχουν φανταστικό μέρος και όντως παρουσιάζουν μία συσχέτιση μεταξύ τους, ειδικά όταν εστιάζουμε σε ένα μικρότερο χρονικό διάστημα όπως κάνουμε στην Εικόνα 21(d) όπου παρατηρούμε ότι τα δείγματα παρουσιάζουν εμφανή συσχέτιση μεταξύ τους.

Στο σημείο αυτό θα πρέπει να σημειώσουμε αυτό που κατά την γνώμη μας προκρίνει την χρήση του λευκού ή του έγχρωμου θορύβου στις προσομοιώσεις μας. Αν δειγματοληπτούμε το σήμα έτσι ώστε να έχουμε ένα δείγμα ανά σύμβολο (οπότε ο ρυθμός δειγματοληψίας είναι $f_s=1/T_s$ όπου T_s είναι η περίοδος των δειγμάτων και αν υποθέσουμε ότι το εύρος ζώνης του δέκτη και του σήματος είναι της τάξης του $1/T_s$ τότε είναι λογικό να υποθέσουμε ότι τα δείγματα του θορύβου από σύμβολο σε σύμβολο είναι ασυσχέτιστα οπότε μπορούμε να χρησιμοποιήσουμε την πιο απλή και πιο γρήγορη *my_awgn*. Ωστόσο στην περίπτωση όπου έχουμε περισσότερα δείγματα ανά σύμβολο τότε η χρήση της *my_awgn* θα σήμαινε ότι το εύρος ζώνης του θορύβου είναι σημαντικά μεγαλύτερο από αυτό του σήματος εξαιτίας των ασυσχέτιστων μεταβολών του μεταξύ των δειγμάτων. Επομένως ο θόρυβος που βρίσκεται εκτός του εύρους ζώνης του σήματος θα μπορούσε να απομακρυνθεί με ένα απλό ηλεκτρικό φίλτρο κάτι που βέβαια δεν ισχύει στην πραγματική περίπτωση. Επομένως η χρήση της *my_awgn* στην περίπτωση αυτή χρήζει ιδιαίτερης προσοχής ώστε να μην οδηγήσει σε υποτίμηση των επιπτώσεων του θορύβου, ειδικά στην περίπτωση όπου χρησιμοποιούμε κάποιου είδους ηλεκτρονικό φίλτρο στον δέκτη.



Εικόνα 21: Υπολογισμός των στατιστικών χαρακτηριστικών των δειγμάτων της *colored_noise*: a) η συνάρτηση αυτοσυσχέτισης, b) η φασματική πυκνότητα ισχύος, c) ένα παράδειγμα δειγμάτων που παράγονται από την συνάρτηση και d) τα ίδια δείγματα σε ένα μικρότερο χρονικό παράθυρο.

7.4 Ο θόρυβος φάσης

Η μοντελοποίηση του θορύβου φάσης ενός οπτικού πομπού είναι ιδιαίτερα σημαντική ειδικά στην περίπτωση της σύμφωνης φώρασης. Όπως δείχνουν οι εξισώσεις (64)-(65) ο θόρυβος φάσης προκαλεί μία στροφή των ρευμάτων i_I και i_Q ως προς τα πεδία e_I και e_Q κατά την γωνία $\Delta\phi$. Η τυχαία μεταβολή της φάσης ϕ μοντελοποιείται με μία τυχαία διαδικασία κίνησης Brown [6], δηλαδή:

$$\Delta\phi = \phi(t_2) - \phi(t_1) = 2\pi \int_{t_1}^{t_2} \mu(t) dt \quad (66)$$

όπου t_2 και t_1 είναι τυχαίες χρονικές στιγμές και $\mu(t)$ είναι μία Gaussian τυχαία διαδικασία με μηδενική μέση τιμή και συνάρτηση αυτοσυσχέτισης,

$$R_{\mu}(t+\tau, t) = \frac{\Delta\nu}{2\pi} \delta(\tau) \quad (67)$$

όπου $\Delta\nu$ το εύρος γραμμής του laser. Μπορούμε εύκολα να δείξουμε ότι,

$$\langle \Delta\phi \rangle = 0 \quad (68)$$

$$\langle \Delta\phi^2 \rangle = 2\pi\Delta\nu(t_2 - t_1) \quad (69)$$

Η (69) μπορεί να χρησιμοποιηθεί για να γεννήσουμε τα στοιχειώδη άλματα της φάσης όταν μεταβαίνουμε από την χρονική στιγμή t_1 στην χρονική στιγμή t_2 . Για την υλοποίηση του θορύβου φάσης χρησιμοποιούμε την συνάρτηση `phase_noise` ενώ ένα παράδειγμα της συνάρτησης αυτής είναι το αρχείο `test_phase_noise`. Στην Εικόνα 22 δείχνουμε ένα παράδειγμα γέννησης θορύβου φάσης από ένα laser με εύρος γραμμής $\Delta\nu=100\text{kHz}$.

phase_noise.m

```
function phi=phase_noise(t,Dv)
% phi=phase_noise(t,Dv)
%
% generation of phase noise samples
% t is the time axis
% Dv is the laser linewidth
%
Dt=[0 diff(t)];
Nt=length(t);

% random jump strength
sm=2*pi*Dv*Dt;

% generate the random jumps
dphi=sm.*randn(1,Nt);

% estimate the phase
phi=cumsum(dphi);
```

test_phase_noise.m

```
% an example for the use of phase_noise
close all
% time window
Tmax=1e-3;
Dv=100e3;
Nt=1e5;
Dt=Tmax/Nt;
t=(1:Nt)*Dt;

% estimate phase
p=phase_noise(t,Dv);

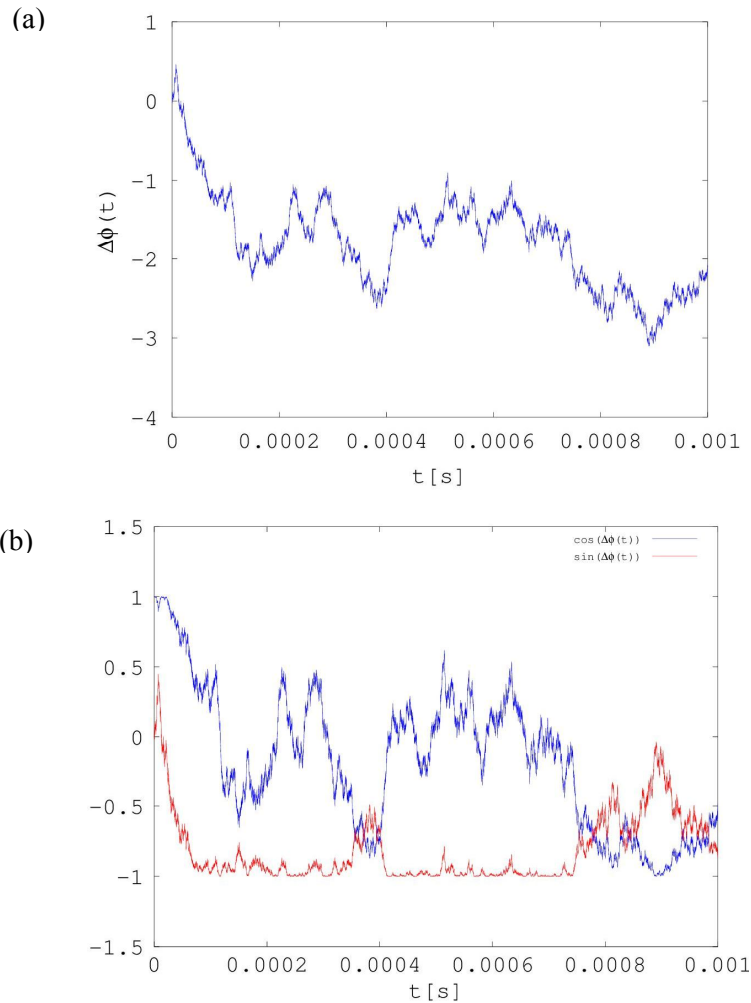
figure(1);
plot(t,p);
```

```

set(gca,'FontSize',20);
xlabel('t[s]');
ylabel('\Delta\phi(t)');

figure(2);
plot(t,cos(p),t,sin(p),'r');
set(gca,'FontSize',20);
xlabel('t[s]');
legend('cos(\Delta\phi(t))','sin(\Delta\phi(t))');
ylim([-1.5 1.5]);

```



Εικόνα 22: (a) παράδειγμα γέννησης τυχαίας φάσης $\Delta\phi$ και (b) οι παράγοντες $\cos(\Delta\phi)$ και $\sin(\Delta\phi)$

7.5 Υπολογισμός των ρευμάτων εξόδου του δέκτη

Το τελικό βήμα είναι να υπολογίσουμε τα φωτορεύματα I/Q όπως αυτά εκφράζονται από τις (64)-(65). Για τον σκοπό αυτό χρησιμοποιούμε την συνάρτηση *iq_currents* την οποία παραθέτουμε παρακάτω. Το αρχείο *example_system* δείχνουμε ένα παράδειγμα χρήσης της *iq_currents* που παράλληλα παρουσιάζει ένα σχεδόν πλήρες μοντέλου του οπτικού συστήματος χρησιμοποιώντας διαμόρφωση 4-QAM. Στην Εικόνα 23 δείχνουμε τους υπολογισμούς που λαμβάνουν χώρα μέσω της *iq_currents*. Έχουμε υποθέσει ότι η λαμβανόμενη ισχύς είναι $P_R = -50\text{dBm}$ (10^{-8}W), ο ρυθμός δειγματοληψίας είναι $R_S = 100\text{MS/s}$ ενώ το εύρος ζώνης του δέκτη θεωρείται ίσο με $B = 300\text{MHz}$. Το άθροισμα του εύρους γραμμής εκπομπής των δύο laser θεωρείται ίσο με 100kHz .

```
iq_currents.m
```

```

function [iI,iQ]=iq_currents(t,E,Dv,pn,receiver);

% calculates the I/Q currents generated at the receiver

```

```

PR=max(E.^2);

% Estimate additive noise PSD
B=receiver.B; % electrical receiver bandwidth
Nf=100; % number of points in the f-axis
fmin=-B; % minimum frequency
fmax=+B; % maximum frequency
Df=(fmax-fmin)/Nf; % frequency spacing
f=fmin:Df:fmax; % frequency axis
Sth=0.25*thermal_psd(abs(f),receiver); % thermal noise psd
Ssh=0.25*shot_psd(abs(f),receiver,PR,pn); % thermal noise psd
Stot=Sth+Ssh; % total noise psd

% estimate noise components
[nI,f1]=colored_noise(t,f,Stot);
[nQ,f2]=colored_noise(t,f,Stot);

% separate real and imaginary field components
eI=real(E);
eQ=imag(E);

% estimate phase noise
dphi=phase_noise(t,Dv);

% estimate I and Q currents
R=receiver.R;
PLO=receiver.PLO;
iI=R*sqrt(PLO)*(eI.*cos(dphi)-eQ.*sin(dphi))+nI;
iQ=R*sqrt(PLO)*(eI.*sin(dphi)+eQ.*cos(dphi))+nQ;

```

```

example_system.m

```

```

% simulation of an example COWS system
%
Rs=1e8; % symbol rate
PT=1e-8; % input power
T=1/Rs; % symbol duration
b=0.5; % Roll off factor
Vpi=3.5; % modulator characteristic voltage
Dv=100e3; % laser linewidth

% Signal and ambient noise parameters
lambda=1.55e-6; % Wavelength
pn=6e10; % ambient light irradiance

% signal modulation
Nsymbols=1000;
smI=Vpi/2*sign(randn(1,Nsymbols)); % symbol amplitudes for I voltage component
smQ=Vpi/2*sign(randn(1,Nsymbols)); % symbol amplitudes for Q voltage component
Nt=10; % points in the t-axis / symbol
Ntotal=(length(smQ)+2)*Nt; % total number of symbols in the t-axis

% time axis
Tmin=-T; % start of the time window
Tmax=Tmin+(length(smQ)+2)*T; % end of the time window
Dt=(Tmax-Tmin)/Ntotal; % time step
t=Tmin+(1:Ntotal)*Dt; % time axis

% Receiver parameters
receiver.RF=10e3; % Feedback resistance
receiver.Cd=20e-12; % photodiode capacitance
receiver.Cds=0.5e-12; % MOS drain/source capacitance
receiver.Cgs=0.5e-12; % MOS gate/source capacitance
receiver.gm=40e-3; % MOS transconductance
receiver.ID=20e-3; % MOS DC drain current
receiver.K=294e-15; % MOS K parameter
receiver.R=0.53; % detector responsivity

```

```

receiver.RD=1000;           % drain resistance
receiver.gamma=1;         % MOS channel noise factor
receiver.B=3/T;           % electrical receiver bandwidth
receiver.PLO=1e-3;        % Receiver LO power
receiver.A2=100;          % amplifier gain
receiver.Dl=40e-9;        % electrical receiver bandwidth
receiver.AR=1e-2;         % receiver area

% Generation of driving voltages
vI=driving_voltage(t,smI,T,b)-Vpi/2;
vQ=driving_voltage(t,smQ,T,b)-Vpi/2;

% Estimation of optical baseband signal
E=iq_modulator(t,vI,vQ,Vpi,PT);

% Estimation of the I/Q currents
[iI,iQ]=iq_currents(t,E,Dv,pn,receiver);

% sample to determine the output symbols
smIo=zeros(1,Nsymbols);   % symbol estimates for the I current component
smQo=Vpi/2*sign(randn(1,Nsymbols)); % symbol estimates for the Q current component

for m=1:Nsymbols
    [j,i]=min(abs(t-m*T));
    smIo(m)=iI(i);
    smQo(m)=iQ(i);
end

% Plot symbols
figure(1);
subplot(211), stem(smI);
set(gca,'FontSize',20);
ylabel('s_I');
subplot(212), stem(smQ,'r'); ylabel('s_Q');
set(gca,'FontSize',20);
ylabel('s_Q');

% Plot I/Q input voltage
figure(2);
subplot(211), plot(t,vI);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_I(t)');
xlim([0 20*T]);
ylim([-1.1*Vpi 0.1*Vpi]);
subplot(212), plot(t,vQ,'r');
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_Q(t)');
ylim([-1.1*Vpi 0.1*Vpi]);
xlim([0 20*T]);

% Plot I/Q output current
figure(3);
subplot(211), plot(t,iI);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_I(t)');
ylim([-1.1*max(abs(iI)) 1.1*max(abs(iI))]);
xlim([0 20*T]);
subplot(212), plot(t,iQ,'r');
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_Q(t)');
ylim([-1.1*max(abs(iQ)) 1.1*max(abs(iQ))]);
xlim([0 20*T]);

```

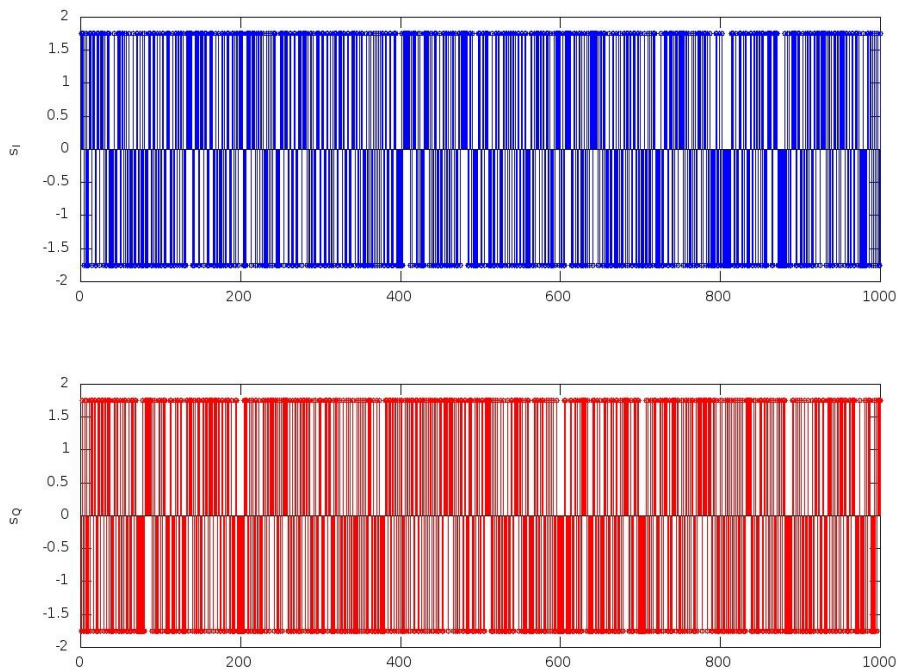
```

% Plot output symbols
figure(4);
subplot(211), stem(smIo);
set(gca,'FontSize',20);
ylabel('s''I');
subplot(212), stem(smQo,'r'); ylabel('\itS\rm_Q');
set(gca,'FontSize',20);
ylabel('s''_Q');

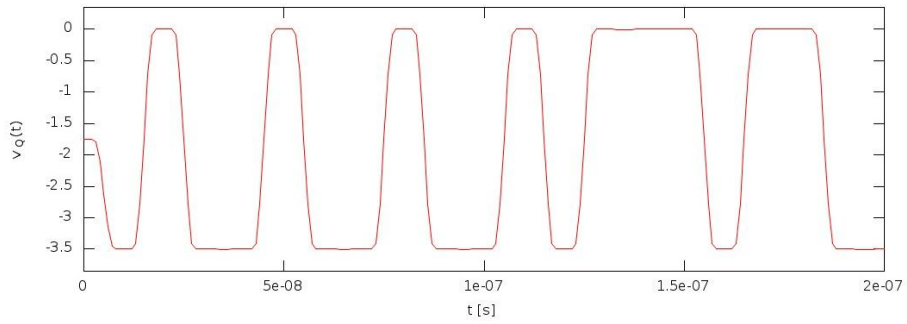
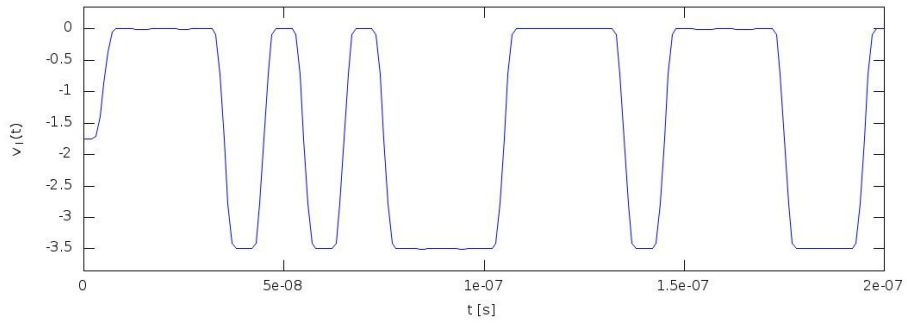
% symbol diagrams
figure(5);
plot(smI,smQ,'o','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor','b'); axis off;
set(gca,'FontSize',20);
title('Input Constellation');
print -color fig23e.jpg
figure(6);
plot(smIo,smQo,'o','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor','b'); axis
off;
set(gca,'FontSize',20);
title('Output Constellation');

```

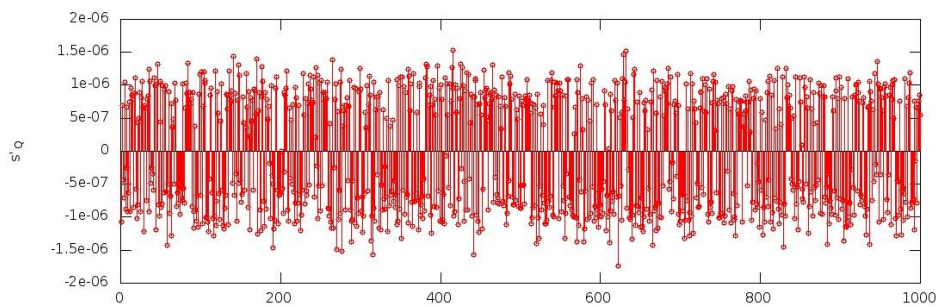
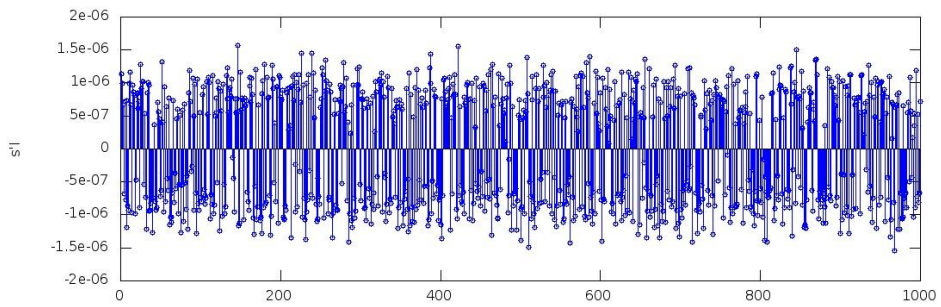
(a)

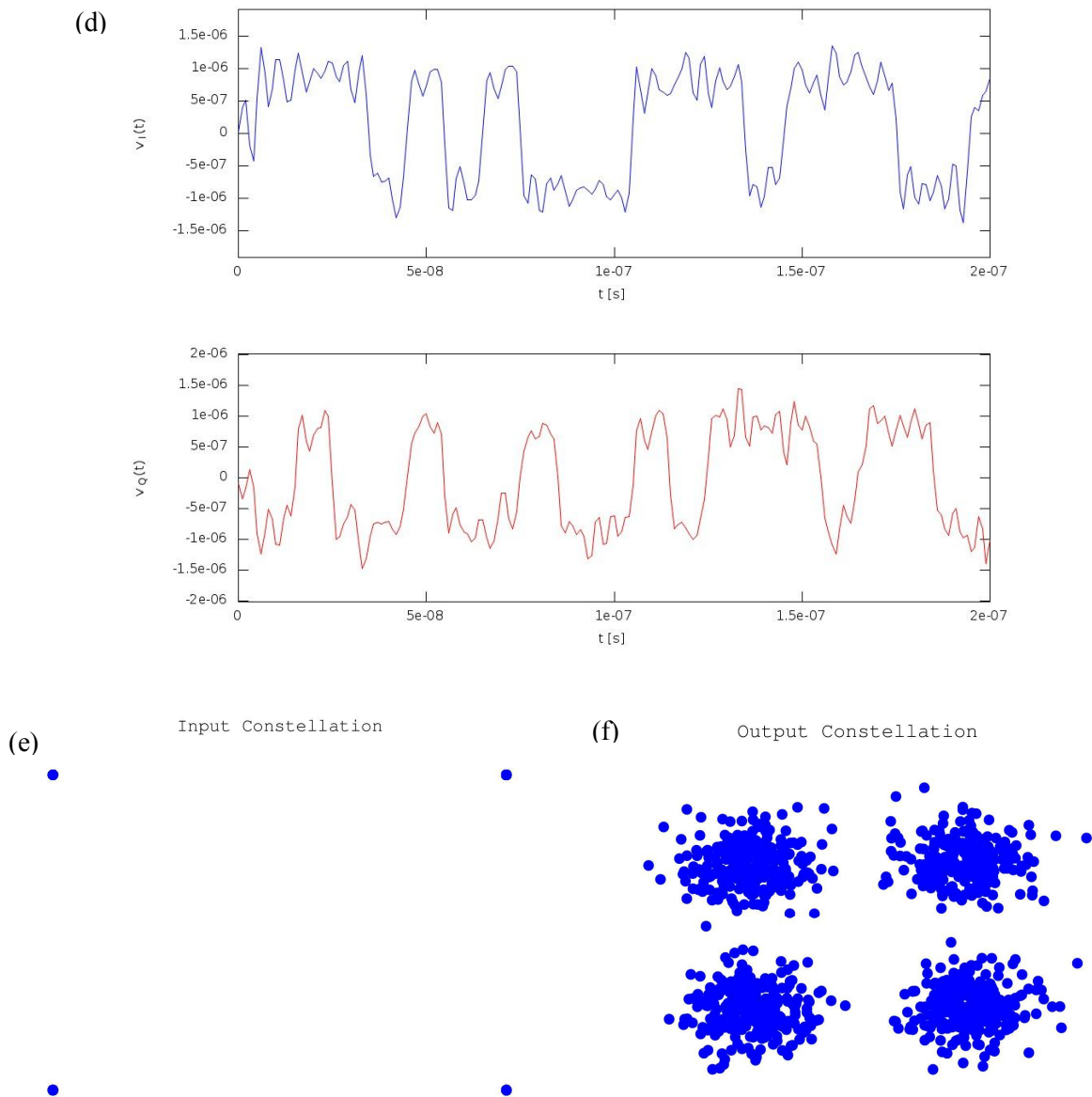


(b)



(c)





Εικόνα 23: Υπολογισμός των στατιστικών χαρακτηριστικών των δειγμάτων της *colored noise*: a) τα σύμβολα εισόδου, b) οι κυματομορφές οδήγησης τάσης, c) τα σύμβολα εξόδου, d) οι κυματομορφές του ρεύματος στην έξοδο στην έξοδο του δέκτη, e) ο αστερισμός συμβόλων στην είσοδο, f) ο αστερισμός συμβόλων στην έξοδο.

8. Μοντελοποίηση του συστήματος από άκρο-σε-άκρο

8.1 Υλοποίηση συνολικού μοντέλου

Το τελικό βήμα είναι να συγκεντρώσουμε ολόκληρο το μοντέλο του συστήματος σε μία συνάρτηση. Η συνάρτηση αυτή είναι η *cows_model* την οποία παραθέτουμε παρακάτω μαζί με την συνάρτηση *cows_plot* η οποία παριστάνει γραφικά την είσοδο και την έξοδο του συστήματος όπως φαίνεται και στην Εικόνα 23. Η συνάρτηση *cows_model* έχει ως είσοδο και ως έξοδο μία μεταβλητή τύπου δομής (structure) του OCTAVE επειδή ο αριθμός των παραμέτρων που χρειάζεται να δοθούν από τον χρήστη είναι πάρα πολύ μεγάλος. Ένα παράδειγμα κλήσης της *cows_model* και της *cows_plot* είναι το αρχείο *cows_example*. Παρακάτω συνοψίζουμε τα πεδία του τύπου που χρησιμοποιείται ως είσοδος και ως έξοδος της *cows_model*:

- cows.Rs: ο ρυθμός των συμβόλων [είσοδος].
- cows.PT: η ισχύς εκπομπής [είσοδος].
- cows.b: η παράμετρος roll-off για τους παλμούς οδήγησης τάσης [είσοδος].
- cows.Vpri: η τάση του διαμορφωτή στην οποία εισάγεται διαφορά φάσης π στο πεδίο εξόδου [είσοδος].
- cows.DvT: το εύρος γραμμής (linewidth) του πομπού [είσοδος].
- cows.lambda: το μήκος κύματος εκπομπής [είσοδος].
- cows.rT: η θέση του πομπού στον τρισδιάστατο χώρο [είσοδος].
- cows.nT: ο προσανατολισμός του δέκτη στο χώρο [είσοδος].
- cows.fi: οι εστιακές αποστάσεις των φακών του πομπού [είσοδος].
- cows.li: οι αποστάσεις μεταξύ των φακών [είσοδος].
- cows.w: το αρχικό εύρος της δέσμης [είσοδος].
- cows.Nsymbols: αριθμός των συμβόλων που μεταδίδονται [είσοδος].
- cows.smI: πλάτη τάσεων εισόδου για τα σύμβολα της I συνιστώσας [είσοδος].
- cows.smQ: πλάτη τάσεων εισόδου για τα σύμβολα της Q συνιστώσας [είσοδος].
- cows.Nt: αριθμός δειγμάτων ανά σύμβολο [είσοδος].
- cows.Ntotal: αριθμός δειγμάτων στον άξονα του χρόνου [είσοδος].
- cows.T: διάρκεια συμβόλου [είσοδος].
- cows.t: άξονας του χρόνου [είσοδος].
- cows.R: αποκρισιμότητα του δέκτη [είσοδος].
- cows.RF: αντίσταση ανάδρασης του ηλεκτρονικού ενισχυτή [είσοδος].
- cows.Cd: χωρητικότητα φωτοδιόδου [είσοδος].
- cows.Cds: χωρητικότητα πηγής/υποδοχής για το FET του δέκτη [είσοδος].
- cows.Cgs: χωρητικότητα πηγής/πύλης για το FET του δέκτη [είσοδος].
- cows.gm: διαγωγιμότητα για το FET του δέκτη [είσοδος].
- cows.ID: ρεύμα πόλωση του FET του δέκτη [είσοδος].
- cows.K: συντελεστής K του FET του δέκτη [είσοδος].
- cows.RD: αντίσταση υποδοχής του FET [είσοδος].

- cows.gamma: παράγοντας θορύβου του FET του δέκτη [είσοδος].
- cows.B: ηλεκτρονικό εύρος ζώνης θορύβου του δέκτη [είσοδος].
- cows.PLO: ισχύς τοπικού ταλαντωτή του δέκτη [είσοδος].
- cows.A2: ενίσχυση του ηλεκτρονικού ενισχυτή του δέκτη [είσοδος].
- cows.DI: εύρος ζώνης του οπτικού φίλτρου [είσοδος].
- cows.AR: επιφάνεια του δέκτη [είσοδος].
- cows.rR: θέση του δέκτη στο χώρο [είσοδος].
- cows.nR: προσανατολισμός του δέκτη στο χώρο [είσοδος].
- cows.DvR: εύρος γραμμής του τοπικού ταλαντωτή [είσοδος].
- cows.pn: φωτεινότητα του θορύβου περιβάλλοντος [είσοδος].
- cows.smIo: πλάτη εξόδου για τα σύμβολα της I συνιστώσας [έξοδος].
- cows.smQo: πλάτη εξόδου για τα σύμβολα της Q συνιστώσας [έξοδος].
- cows.vI: τάση οδήγησης για την I συνιστώσα [έξοδος].
- cows.vQ: τάση οδήγησης για την Q συνιστώσα [έξοδος].
- cows.iI: ρεύμα εξόδου για την I συνιστώσα [έξοδος].
- cows.iQ: ρεύμα εξόδου για την Q συνιστώσα [έξοδος].
- cows.PR: η ισχύς λήψης [έξοδος].

Στην παραπάνω λίστα έχουμε σημειώσει ποιες παράμετροι χρειάζονται ως είσοδοι κατά την κλήση του `cows_model` και τα δεδομένα εξόδου που προκύπτουν από την κλήση του.

cows_model.m

```
function cows_o=cows_model(cows);

% TX parameters
Rs=cows.Rs; % symbol rate
PT=cows.PT; % transmit power
b=cows.b; % Roll off transmitter pulse factor
Vpi=cows.Vpi; % modulator characteristic voltage
DvT=cows.DvT; % transmitter laser linewidth
lambda=cows.lambda; % laser wavelength
rT=cows.rT; % transmitter position
nT=cows.nT; % transmitter orientation

% TX optics
fi=cows.fi; % focal distance of the lenses
li=cows.li; % distances between the lenses
w=cows.w; % initial beamwidth

% signal parameters
Nsymbols=cows.Nsymbols; % number of symbols transmitted
smI=cows.smI; % symbol voltage amplitudes for I component
smQ=cows.smQ; % symbol voltage amplitudes for Q component
Nt=cows.Nt; % points in the t-axis / symbol
Ntotal=cows.Ntotal; % total number of symbols in the t-axis

% time axis
T=cows.T; % symbol duration
t=cows.t; % time axis

% Receiver parameters
RF=cows.RF; % Feedback resistance
Cd=cows.Cd; % photodiode capacitance
```

```

Cds=cows.Cds;           % MOS drain/source capacitance
Cgs=cows.Cgs;           % MOS gate/source capacitance
gm=cows.gm;             % MOS transconductance
ID=cows.ID;             % MOS DC drain current
K=cows.K;               % MOS K parameter
R=cows.R;               % detector responsivity
RD=cows.RD;            % drain resistance
gamma=cows.gamma;      % MOS channel noise factor
B=cows.B;               % electrical receiver bandwidth
PLO=cows.PLO;          % Receiver LO power
A2=cows.A2;             % amplifier gain
Dl=cows.Dl;            % electrical receiver bandwidth
AR=cows.AR;            % receiver area
rR=cows.rR;            % receiver position
nR=cows.nR;            % receiver orientation
DvR=cows.DvR;          % LO linewidth
pn=cows.pn;            % ambient light noise irradiance

Dv=cows.DvR+cows.DvT;   % laser linewidth

% assign receiver parameters

% Receiver parameters
receiver.RF=RF;          % Feedback resistance
receiver.Cd=Cd;          % photodiode capacitance
receiver.Cds=Cds;        % MOS drain/source capacitance
receiver.Cgs=Cgs;        % MOS gate/source capacitance
receiver.gm=gm;          % MOS transconductance
receiver.ID=ID;          % MOS DC drain current
receiver.K=K;            % MOS K parameter
receiver.R=R;            % detector responsivity
receiver.RD=RD;          % drain resistance
receiver.gamma=gamma;    % MOS channel noise factor
receiver.B=B;            % electrical receiver bandwidth
receiver.PLO=PLO;        % Receiver LO power
receiver.A2=A2;          % amplifier gain
receiver.Dl=Dl;          % electrical receiver bandwidth
receiver.AR=AR;          % receiver area

% Duplicate model parameters
cows_o=cows;

% Calculate power budget
PR=power_budget(rT,nT,PT,fi,li,lambda,w,rR,nR,AR);
cows_o.PR=PR;

% Generation of driving voltages
vI=driving_voltage(t,smI,T,b)-Vpi/2;
vQ=driving_voltage(t,smQ,T,b)-Vpi/2;
cows_o.vI=vI;
cows_o.vQ=vQ;

% Estimation of optical baseband signal at the transmitter
E=iq_modulator(t,vI,vQ,Vpi,PT);
cows_o.E=E;
ER=E.*sqrt(PR/PT);
cows_o.ER=ER;

% Estimation of the I/Q currents
[iI,iQ]=iq_currents(t,ER,Dv,pn,receiver);
cows_o.iI=iI;
cows_o.iQ=iQ;

% sample to determine the output symbols
smIo=zeros(1,Nsymbols); % symbol estimates for the I current component
smQo=zeros(1,Nsymbols); % symbol estimates for the Q current component

for m=1:Nsymbols

```

```

    [j,i]=min(abs(t-m*T));
    smIo(m)=iI(i);
    smQo(m)=iQ(i);
end
cows_o.smIo=smIo;
cows_o.smQo=smQo;

```

cows_plot.m

```

function cows_plot(cows);

% initial symbols
smI=cows.smI;
smQ=cows.smQ;

% final symbols
smIo=cows.smIo;
smQo=cows.smQo;

% input voltages
vI=cows.vI;
vQ=cows.vQ;

% output currents
iI=cows.iI;
iQ=cows.iQ;

% time axis
t=cows.t;
T=cows.T;

% modulator pi-voltage
Vpi=cows.Vpi;

% Plot symbols
figure(1);
subplot(211), stem(smI);
ylabel('s_I');
subplot(212), stem(smQ,'r'); ylabel('S_Q');
ylabel('s_Q');

% Plot I/Q input voltage
figure(2);
subplot(211), plot(t,vI);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_I(t)');
xlim([0 20*T]);
ylim([-1.1*Vpi 0.1*Vpi]);
subplot(212), plot(t,vQ,'r');
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('v_Q(t)');
ylim([-1.1*Vpi 0.1*Vpi]);
xlim([0 20*T]);

% Plot I/Q output current
figure(3);
subplot(211), plot(t,iI);
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('i_I(t)');
ylim([-1.1*max(abs(iI)) 1.1*max(abs(iI))]);
xlim([0 20*T]);
subplot(212), plot(t,iQ,'r');
set(gca,'FontSize',20);
xlabel('t [s]');
ylabel('i_Q(t)');

```

```

ylim([-1.1*max(abs(iQ)) 1.1*max(abs(iQ))]);
xlim([0 20*T]);

% Plot output symbols
figure(4);
subplot(211), stem(smIo);
set(gca,'FontSize',20);
ylabel('s''_I');
subplot(212), stem(smQo,'r'); ylabel('S_Q');
set(gca,'FontSize',20);
ylabel('s''_Q');

% symbol diagrams
figure(5);
plot(smI,smQ,'o','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor','b'); axis off;
set(gca,'FontSize',20);
title('Input Constellation');
print -color fig25e.jpg
figure(6);
plot(smIo,smQo,'o','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor','b'); axis
off;
set(gca,'FontSize',20);
title('Output Constellation');

```

cows_example.m

```

% Transmitter parameters
cows.Rs=1e9; % symbol rate
cows.PT=1e-3; % transmit power
cows.b=0.5; % Roll off transmitter pulse factor
cows.Vpi=3.5; % modulator characteristic voltage
cows.DvT=300e3; % transmitter laser linewidth
cows.lambda=1.55e-6; % laser wavelength
cows.rT=[0 0 5]; % transmitter position
cows.nT=[0 0 -1]; % transmitter orientation
cows.fi=cell(1,1); % transmitter lens focal lengths
cows.li=cell(1,1); % transmitter lens distances
cows.w=2e-6; % initial beam width

% signal parameters
cows.Nsymbols=1000; % number of symbols transmitted
smI1=sign(randn(1,cows.Nsymbols)); % symbols for I component
cows.smI=cows.Vpi/2*smI1; % symbol voltage amplitudes for I component
smQ1=sign(randn(1,cows.Nsymbols)); % symbols for Q component
cows.smQ=cows.Vpi/2*smQ1; % symbol voltage amplitudes for Q component
cows.Nt=10; % points in the t-axis / symbol
cows.Ntotal=(length(cows.smQ)+2)*cows.Nt; % total number of symbols in the t-axis

% time axis
cows.T=1/cows.Rs; % symbol duration
Tmin=-cows.T; % start of the time window
Tmax=Tmin+(length(cows.smQ)+2)*cows.T; % end of the time window
Dt=(Tmax-Tmin)/cows.Ntotal; % time step
cows.t=Tmin+(1:cows.Ntotal)*Dt; % time axis

% Receiver parameters
cows.RF=10e3; % Feedback resistance
cows.Cd=20e-12; % photodiode capacitance
cows.Cds=0.5e-12; % MOS drain/source capacitance
cows.Cgs=0.5e-12; % MOS gate/source capacitance
cows.gm=40e-3; % MOS transconductance
cows.ID=20e-3; % MOS DC drain current
cows.K=294e-15; % MOS K parameter
cows.R=0.53; % detector responsivity
cows.RD=1000; % drain resistance

```

```

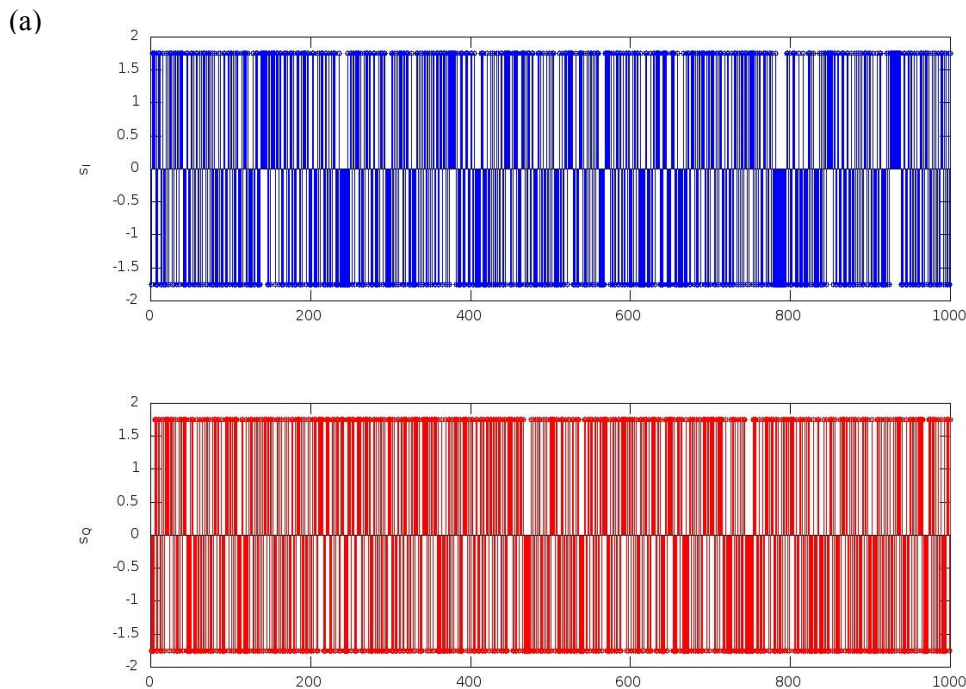
cows.gamma=1; % MOS channel noise factor
cows.B=3/cows.T; % electrical receiver bandwidth
cows.PLO=2e-3; % Receiver LO power
cows.A2=100; % amplifier gain
cows.Dl=40e-9; % electrical receiver bandwidth
cows.AR=1e-2; % receiver area
cows.rR=[0 0 0]; % receiver position
cows.nR=[0 0 1]; % receiver orientation
cows.DvR=100e3; % LO linewidth
cows.pn=6e10; % ambient light noise irradiance

cows_o=cows_model(cows);
cows_plot(cows_o);

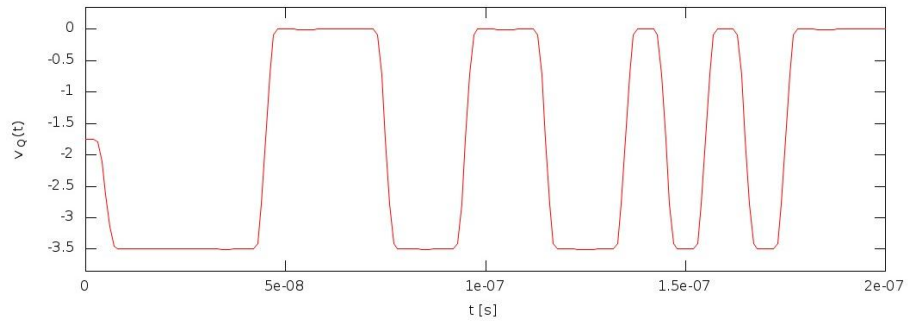
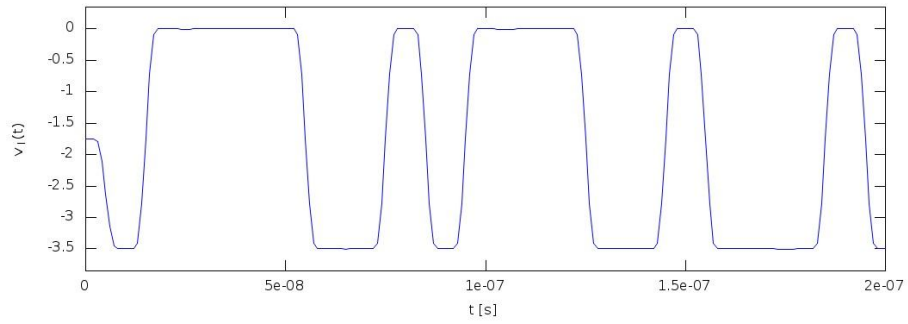
```

8.2 Παραδείγματα

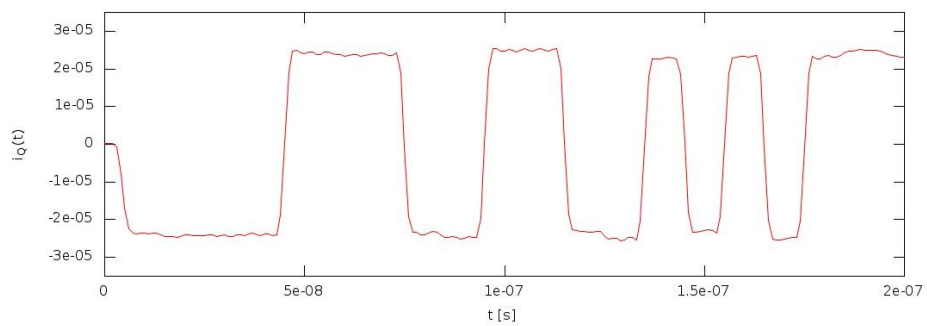
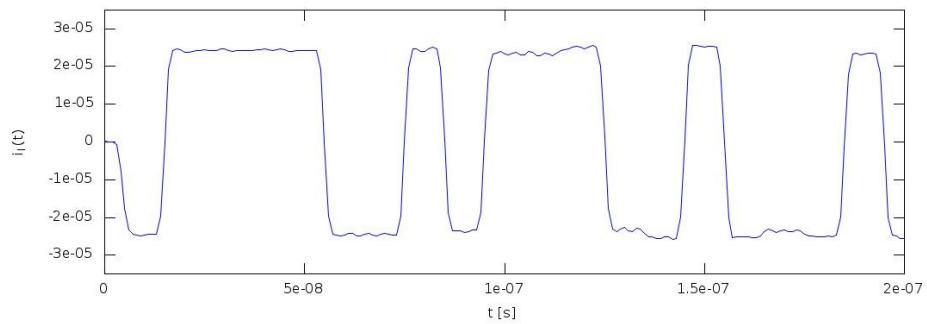
Στην ενότητα αυτή θα παρουσιάσουμε μερικά παραδείγματα χρήσης του μοντέλου για διάφορες τιμές των παραμέτρων του συστήματος. Στις επόμενες προσομοιώσεις, εκτός αν αναφέρουμε κάτι διαφορετικό οι τιμές των παραμέτρων του ηλεκτρονικού ενισχυτή του δέκτη (C_{gs} , C_{ds} , κτλ) θα είναι αυτές που αναφέρονται στο Π1.1. Ο δέκτης θεωρούμε ότι βρίσκεται τοποθετημένος στο σημείο (0,0,0) ενώ ο πομπός βρίσκεται πάνω από τον δέκτη σε απόσταση 5m. Ο πομπός και ο δέκτης είναι προσανατολισμένοι ο ένας προς τον άλλο και η ισχύς εκπομπής είναι $P_T=1\text{mW}$ ενώ η ισχύς του τοπικού ταλαντωτή είναι $P_{LO}=2\text{mW}$. Το εύρος γραμμής του τοπικού ταλαντωτή και του πομπού θεωρούνται ίσα με 100kHz και 300kHz αντίστοιχα. Αρχικά θέτουμε τον ρυθμό συμβόλων ίσο με 100MS/s και θεωρούμε 4-QAM διαμόρφωση. Η ισχύς λήψης στην περίπτωση αυτή είναι $P_R=4\mu\text{W}$. Παρατηρούμε ότι ο προσθετικός θόρυβος δεν έχει μεγάλη επίδραση στο σήμα και πως ο αστερισμός εξόδου παρουσιάζεται στραμμένος εξαιτίας του θορύβου φάσης. Οι επιδόσεις του συστήματος φαίνονται στην Εικόνα 24. Στην Εικόνα 25 δίνουμε τα αποτελέσματα στην περίπτωση όπου ο ρυθμός σηματοδοσίας είναι 1GS/s.

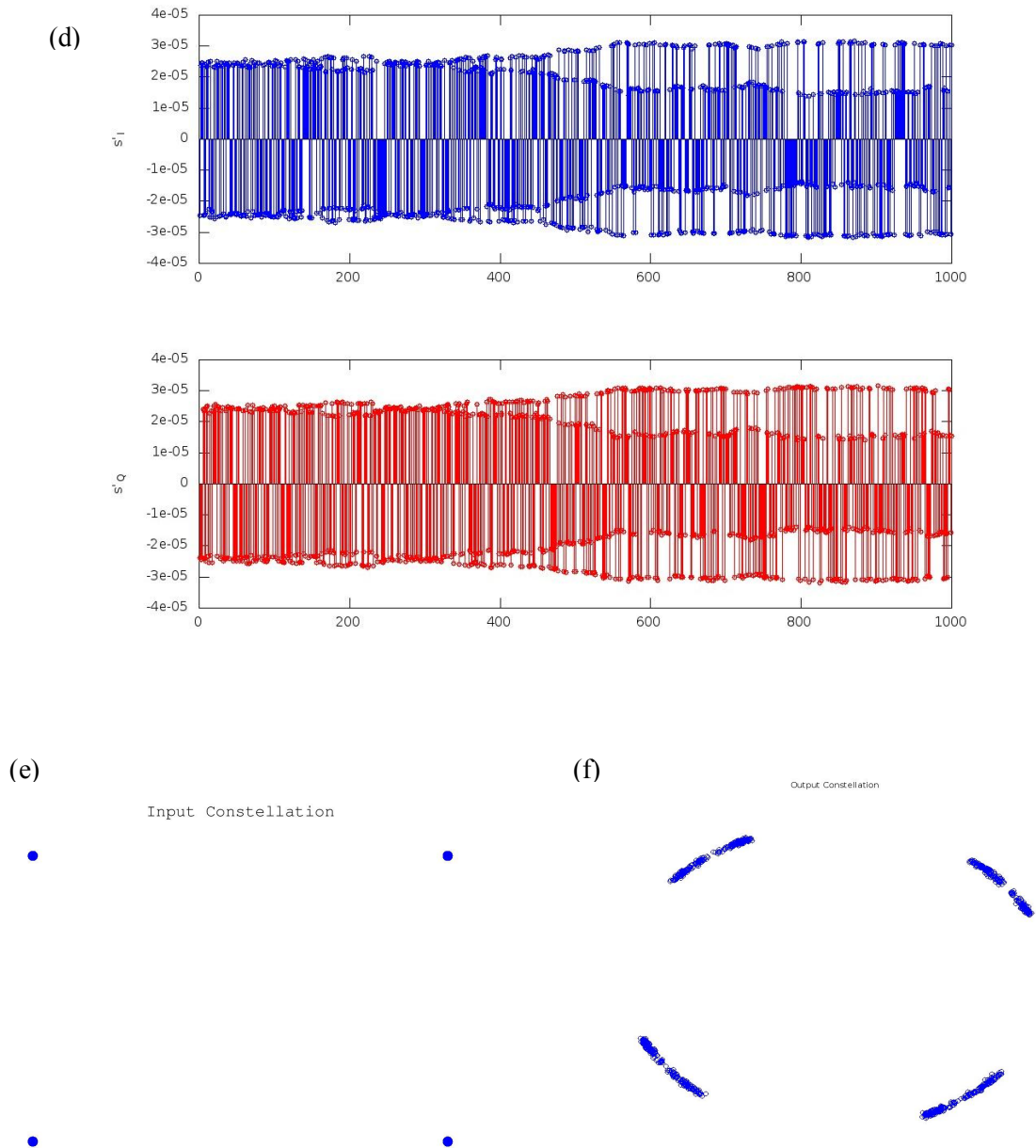


(b)



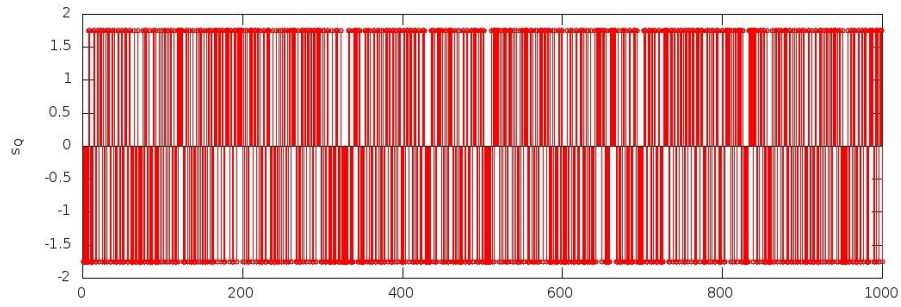
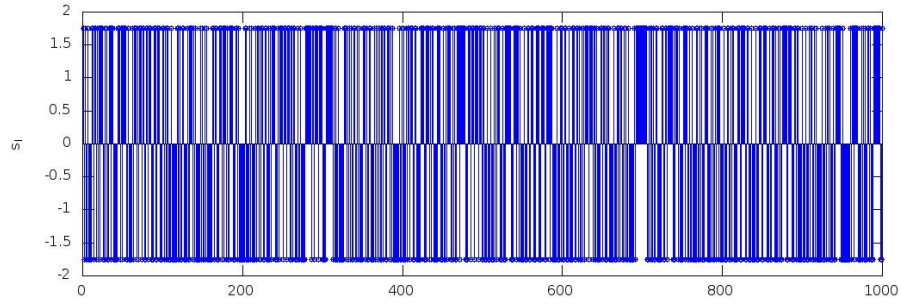
(c)



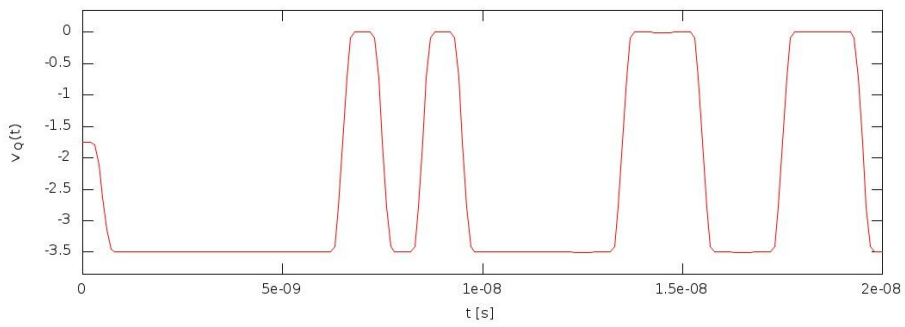
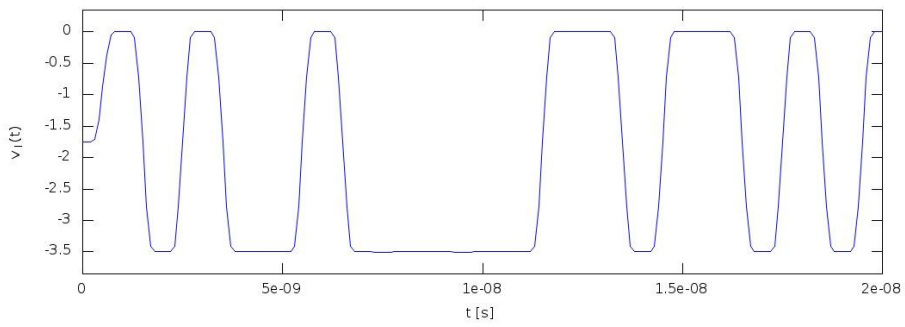


Εικόνα 24: Υπολογισμός των στατιστικών χαρακτηριστικών των δειγμάτων της *colored_noise*: a) τα σύμβολα εισόδου, b) οι κυματομορφές οδήγησης τάσης, c) οι κυματομορφές του ρεύματος στην έξοδο στην έξοδο του δέκτη, d) τα σύμβολα εξόδου e) ο αστερισμός συμβόλων στην είσοδο, f) ο αστερισμός συμβόλων στην έξοδο θεωρώντας ότι ο ρυθμός σηματοδοσίας είναι $R_S=100\text{MS/s}$.

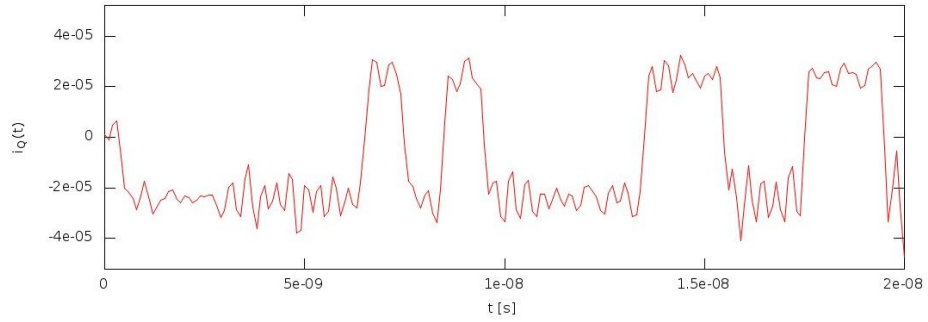
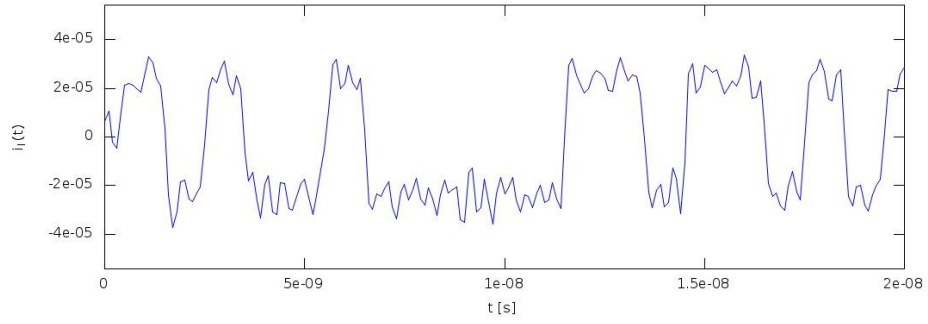
(a)



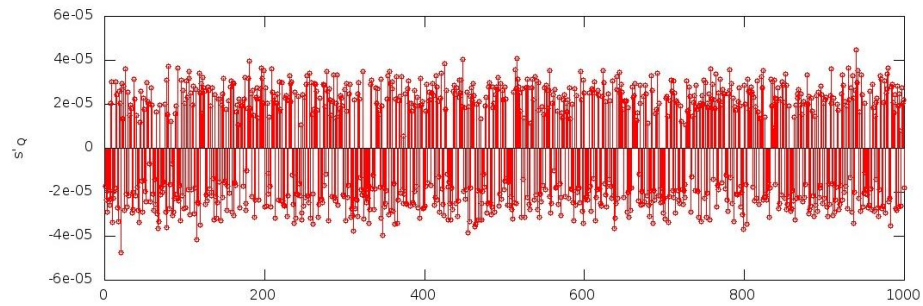
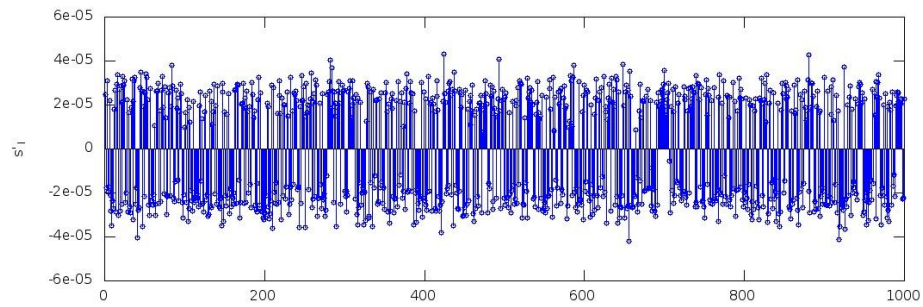
(b)

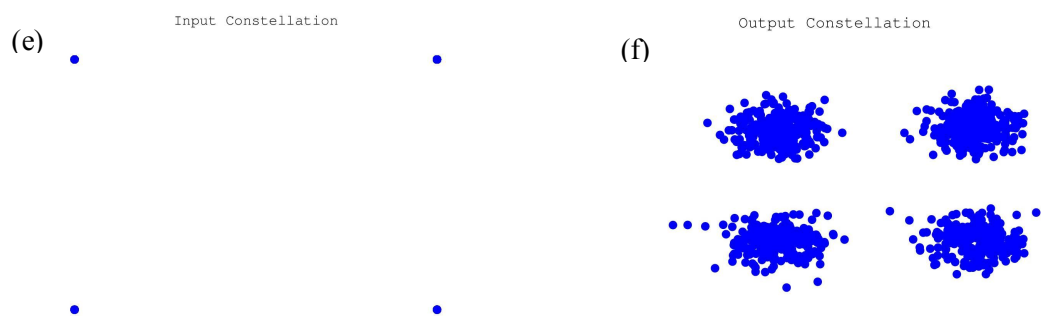


(c)



(d)





Εικόνα 25: Υπολογισμός των στατιστικών χαρακτηριστικών των δειγμάτων της *colored_noise*: a) τα σύμβολα εισόδου, b) οι κυματομορφές οδήγησης τάσης, c) οι κυματομορφές του ρεύματος στην έξοδο στην έξοδο του δέκτη, d) τα σύμβολα εξόδου e) ο αστερισμός συμβόλων στην είσοδο, f) ο αστερισμός συμβόλων στην έξοδο θεωρώντας ότι ο ρυθμός σηματοδότησης είναι $R_s=1\text{GS/s}$.

9. Συμπεράσματα

Στο παραδοτέο αυτό υλοποιήσαμε το μοντέλο του συστήματος όπως αυτό φαίνεται στην Εικόνα 2. Εξηγήσαμε την υλοποίηση του διαμορφωτή, της διάδοσης της δέσμης του πομπού και την υλοποίηση του δέκτη. Το μοντέλο μπορεί να χρησιμοποιηθεί για την μελέτη των μονοδιάστατων ή διδιάστατων συστημάτων διαμόρφωσης όπως το PSK, το QAM και το DPSK. Εμπεριέχει τον προσθετικό θόρυβο, το και το θόρυβο φάσης και υπολογίζει τις εκτιμήσεις των συμβόλων στην έξοδο του δέκτη.

10. Αναφορές

- [1] Το GNU-OCTAVE είναι ελεύθερα διαθέσιμο στην διεύθυνση <https://www.gnu.org/software/octave/>
- [2] «Λεπτομέρειες μοντελοποίησης οπτικού σύμφωνου καναλιού με εργαλεία ανοιχτού λογισμικού», παραδοτέο Π1.1. του προγράμματος «Σύμφωνα Ασύρματα Οπτικά Συστήματα ως Δομικά Στοιχεία για την Υλοποίηση Δικτύων Οικίας και Επαγγελματικών Χώρων Επόμενης Γενιάς, (COWS)», διαθέσιμο ηλεκτρονικά στην διεύθυνση:
- [3] K.-P. Ho and J. M. Kahn “Spectrum of externally modulated optical signals”, J. Lightw. Technol., 22(2):658–663, Feb. 2004.
- [4] P. A. Bélanger, “Beam propagation and the ABCD ray matrices”, Opt. Lett. 16 (4), 196 (1991)
- [5] J.M. Kahn and J. R. Barry, "Wireless Infrared Communications", Proc. of the IEEE, vol. 85, pp. 265-298 (1997).
- [6] Einarsson, G., Strandberg, J., Monroy, I.T.: “Error probability evaluation of optical systems disturbed by phase noise and additive noise”, IEEE J. Lightwave Technol. 13, 1847–1852 (1995).