جامعة دمشق

كلية الهندسة الميكانيكية والكهربائية

قسم الالكترونيات والاتصالات

كشف الاختراقات في شبكات الألياف الضوئية

مشروع أعد لنيل درجة الماجستير في هندسة الالكترونيات والاتصالات

إعداد الطالب

م. محمد نصر

بإشراف

أ.د.م. نديم شاهين

العام الدراسي

2011

كلمة شكر

الشكرُ جزيل الشكر وخالص التقدير لصناع أجيال هذه الأمة العظيمة، وبالأخص لجامعتنا العزيزة وكوادرها الفنية والتعليمية التي ساهمت في إنجاز هذا العمل وإخراجه إلى النور.

وللأستاذ الدكتور المهندس نديم شاهين أقدم الشكر الأكبر لمساعدته الكبيرة، وإشرافه ونصائحه القيمة لعملي هذا.

الإهداء

إلى جميع من ساهم في دعمي خلال مسيرتي الدراسية ....

فهرس المحتويات

الملخص 10
مقدمة 11
الفصل الأول : الألياف الضوئية 12
1.1. طبيعة الضوء:..... 13
1.1.1. الضوء كموجة كهرطيسية: 13
1.2. إرسال الضوء عبر الليف الضوئي:..... 15
1.2.1. خصائص الزجاج:..... 17
1.3. مبدأ عمل الليف الضوئي : 20
1.3.1. انتشار الضوء ضمن الليف :..... 21
الفصل الثاني : أمن الشبكات Network Security 23
1.1. الشبكات اللاسلكية Wireless Network : 24
1.2. الشبكات السلكية Wired Network:..... 24
1.3. شبكات الألياف الضوئية Fiber optic Network : 25
1.1.3 . وصلة مباشرة: 25
2.1.3. الاختراق بثني الليف: 26
3.1.3. الطريقة الكمومية : 27
1.4. حماية الشبكات من الاختراق: 28
الفصل الثالث : المبادئ العامة في القياسات الضوئية 29
3.1. القياس باستخدام جهاز الانعكاس:..... 30
3.2. مبادئ القياس ثنائي الاتجاه :..... 32
3.3. مبادئ تحليل الطيف : 33
3.4. مبادئ قياس تشتت القطبية: 34
3.5. مبادئ قياس التشتت اللوني :..... 36
3.6. مبادئ قياس الاستطاعة الضوئية والضياعات OFI : 39
3.7. معايير الألياف الضوئية :..... 40
الفصل الرابع :الجزء العملي..... 41
1.1. بنية نظام الحماية:..... 42

1.2. التنفيذ العملي: 43

1.3. واجهة البرنامج:45

الخاتمة و الآفاق المستقبلية.....52

الملاحق.....53

الملحق 1 : برنامج المحاكاة 53

الملحق 2 المحاكاة Simulation.....80

المراجع.....95

فهرس الأشكال

الشكل(1-1).بنية الموجة الكهرطيسية 14

الشكل(1-2). الاستقطاب الدائري لموجة كهرطيسية 14

الشكل(1-3). تجربة يونغ 15

الشكل(1-4). بنية الليف الضوئي 16

الشكل(1-5).مخطط امتصاص الضوء تبعاً لطول الموجة 18

الشكل(1-6).النوافذ المستخدمة لأطوال الموجة 18

الشكل(1-7) أنماط العمل الموجودة في الألياف الضوئية 20

الشكل(1-8). ليف متعدد الأنماط بقفزة قرينة الإنكسار 20

الشكل(1-9). ليف متعدد الأنماط بتدرج قرينة الإنكسار: 21

الشكل(1-10). الليف وحيد النمط 21

الشكل(2-1).أدوات الاختراق لتنفيذ وصلة مباشرة 26

الشكل(2-2). ثني الليف **28**

الشكل(3-1).جهاز الانعكاس 30

الشكل(3-2).الضياعات في الأنماط المختلفة لليف 32

الشكل(3-3).قياسات محلل الطيف 34

الشكل(3-4).انعكاس القطبية 35

الشكل(3-5).طريقة قياس PMD 35

الشكل(3-6).معايير ITU-T 37

الشكل(3-7).تأخير الموجة 38

الشكل(3-8).منحني التأخير 39

الشكل(3-9).قياس الاستطاعة 39

الشكل(4-3).المخطط الصندوقي لنظام كشف الاختراق 43

الشكل(4-4). بنية نظام اتصالات ضوئي بسيط 44

الشكل(4-5). معدل الخطأ في الحالة المثالية 44

الشكل(4-6).المخطط الضندوقي لنظام الاتصالات في حالة وجود ضجيج 45

الشكل(4-7). واجهة المحاكاة 45

الشكل(4-8).التمثيل الفراغي لتوزع الاستطاعة 46

الشكل(4-9).توزع الاستطاعة في حال وجود اختراق 46

الشكل(4-10).الاستطاعة المستقبلة في حالة العمل الطبيعي 47

الشكل(4-11).الاستطاعة المستقبلة في حال وجود اختراق 48

الشكل(4-12).منحي معدل الخطأ بدلالة نسبة الشارة للضجيج 49

الشكل(4–13).واجهة البرنامج باستخدام لغة #c 50
الشكل التالي(4–14).قياس المسافة باستخدام جهازOTDR 51

فهرس المصطلحات العلمية

المصطلح	المعنى
Attenuation	تخامد
Bandwidth	سعة القناة
Electromagnetic Wave	موجة كهرطيسية
Single-mode Fiber	ليف وحيد النمط
Multi-mode Fiber	ليف متعدد الأنماط
Shield	التحجيب
Signal-noise ratio	نسبة الإشارة للضجيج
BER	نسبة الخطأ
OTDR	جهاز قياس الاستطاعة الضوئية المنعكسة
Numeric Aperture	الفتحة العددية
Cipher	تشفير
Critical angle	الزاوية الحرجة
Dispersion	التشتت
FIBER OPTICS	الألياف الضوئية
Fiber optic Network	شبكة الالياف الضوئية
Frequency Hopping	القفز الترددي
Heterojunction	الوصلة الثنائية نصف الناقلة
Interference	التداخل
LASER	الضوء الليزر
Light Source	منبع الضوء
Long Wave	موجة طويلة
Medium Wave	موجة متوسطة
Mie	تبعثر Mie
NETWORK SECURITY	أمن الشبكات
Noise	الضجيج
optical spectrum analyzer	محلل الطيف الضوئي

كاشف ضوئي	**Photo Detector**
الفوتون	**Photon**
ترانزستور ضوئي	**Photo Transistor**
الاستقطاب	**Polarization**
تبعثر رايلي	**Rayleigh**
جهاز قياس الانعكاس	**Reflectometer**
موجة قصيرة	**Shortwave**
تقنية التقسيم الموجي	**wavelength division multiplex**
لاسلكي	**Wireless**

الملخص

تعتبر الألياف الضوئية من أهم تقنيات الاتصالات السلكية لما تتمتع به من مزايا كثيرة وأهمها:

- سعة قناة كبيرة.
- تخامد صغير جداً.
- سرعة نقل عالية جداً.
- ممانعة للتداخل.

لذلك أصبحت شائعة الاستخدام في التطبيقات المدنية والعسكرية على حد سواء. نناقش في هذه الأطروحة أحد أهم المواضيع المتعلقة بالاتصالات ألا وهو أمن الألياف الضوئية، ونقترح بنية لنظام كشف وتحديد أماكن الاختراق فيها.

مقدمة

شهدت الاتصالات في القرن الماضي تطوراً كبيراً وكان الدافع الأساسي لهذا التطور الحاجات العسكرية ، حيث شهد هذا القرن الحربين العالميتين ولعبت الاتصالات دوراً هاماً فيهما وكذلك في باقي الحروب مما دفع إلى البحث عن تقنيات متعددة ذات تطبيقات عسكرية وأمنية مثل التشفير وتقنيات الطيف المنثور،وذلك من أجل تأمين وصول المعلومة بشكل آمن وغير قابل للتشويش، هذه الحاجة دفعت باتجاه تأمين وسط اتصال (قناة اتصال) أكثر أمناً مما أدى إلى البحث في موضوع الألياف الضوئية .

عند بدايات العمل بالألياف الضوئية كانت تعاني من تخامد عال ولم تستخدم على نطاق واسع لكن ثمانينات القرن شهدت تطوراً كبيراً فظهرت ألياف بتخامد منخفض مما سمح باستخدامها بكثرة بالإضافة إلى سعة النقل الكبيرة مقارنة بباقي وسائل الاتصالات التي كانت مستخدمة (لاسلكية، سلكية باستخدام الكوابل المحورية، إلخ ...).

دخلت الألياف الضوئية في كثير من التطبيقات حيث استخدمت في هياكل الطائرات المدنية لقياس مقدار التشوه الحاصل على الهيكل، كما استخدمت في بعض المطارات العالمية في مجال تأمين المطار ومنع الاختراق.

نناقش في هذا البحث موضوعاً غاية في الأهمية وهو كشف الاختراق في الألياف الضوئية ونقترح بنية نظام لكشف الاختراق، ولهذا البحث الكثير من التطبيقات التي تقع أغلبها في المجال الأمني ونذكر منها:

1. حماية الأماكن الحساسة وذات الأهمية الكبيرة مثل البنوك ومقرات القيادة.
2. حماية شبكات الاتصالات العسكرية من الاختراق والتجسس.

تناقش هذه الأطروحة في البداية موضوع الألياف الضوئية من حيث طريقة انتشار الضوء وأنماط انتشاره ثم تناقش موضوع التعديل وتعدد المستخدمين وفي الفصل الأخير نقدم بنية النظام المقترحة لكشف الاختراق وتحديد موقعه وإجراء محاكاة للنظام باستخدام بيئة الماتلاب وبيئة net. ونعرض النتائج التي تم الحصول عليها

الفصل الأول : الألياف الضوئية

FIBER OPTICS

1.1. طبيعة الضوء :[15],[7],[6],[1]

يمكن التعبير عن الضوء بثلاث طرق:

I. الأشعة: وهذا التعبير من الفيزياء التقليدية حيث يمكن اعتبار الضوء كمجموعة من الأشعة التي يمكن لها أن تنعكس أو تنكسر. هذا الشرح مفيد لتبيسط مفهوم الضوء لكنه لا يقدم شرحاً للظواهر التي تحدث في الإتصالات الضوئية.

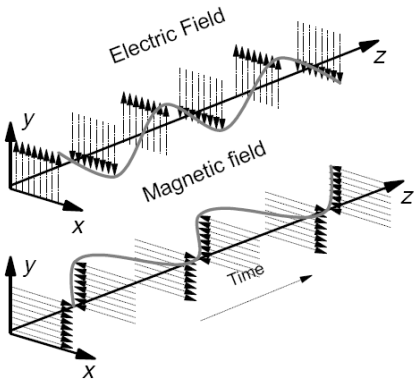
II. موجة كهرطيسية: وهو الشرح الأكثر استخداماً في الإتصالات لتوضيح طبيعة الضوء حيث يمكن اعتباره كموجة كهرطيسية، وبهذا فإن الضوء لا يختلف عن الأمواج الراديوية سوى أنه بطول موجة أقصر (تردد أعلى).

III. فوتونات: وفيه ينظر إلى الضوء على أنه مجموعة من الجزيئات الصغيرة جداً تسمى فوتونات، هذا الشرح يقدم تفسيرات لظوهر تعجز الأمواج الكهرطيسية عن تفسيرها كالأثر الكهربائي حيث يؤدي سقوط فوتون على مادة إلى تحرر إلكترون وبالتالي تشكل تيار كهربائي.

مما سبق نجد أنه يمكن التعبير عن الضوء كموجة كهرطيسية عند دراسة الإنتشار والنظر إليه على أنه فوتونات عند دراسة الأثر المتبادل بين الضوء ومادة ما.

1.1.1. الضوء كموجة كهرطيسية:

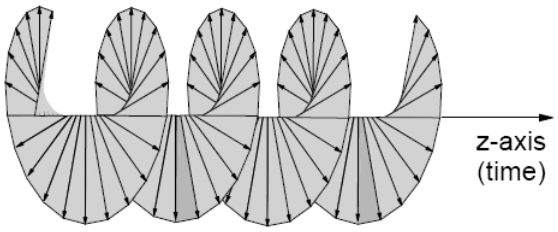
من أكثر الطرق المستخدمة في تفسير الضوء اعتباره كموجة راديوية يختلف عنها بطول الموجة . تتكون الموجة الكهرطيسية كما نعلم من حقلين: حقل كهربائي وحقل مغناطيسي متعامدين مع بعضهما البعض، ويمكن لهذه الموجة أن تنتقل بسرعة الضوء باتجاه عمودي على المستوي المشكل من الحقلين. عند تمثيل الموجة في الفراغ الثلاثي غالباً ما يكون الحقل الكهربائي موازٍ للمحور Y والحقل المغناطيسي موازٍ للمحور X واتجاه الإنتشار هو المحور Z. يوضح الشكل (1-1) بنية الموجة الكهرطيسية.



الشكل(1−1).بنية الموجة الكهرطيسية

الاستقطاب Polarization :

يعرف الاستقطاب بأنه اتجاه اهتزاز الحقل الكهربائي بالنسبة لجهة الإنتشار . في الشكل السابق وجهنا الحقل الكهربائي بالمستوي العمودي والمغناطيسي في المستوي الأفقي، يمكننا تغيير هذا التوجيه بجعل الحقل الكهربائي في المستوي الأفقي والحقل المغناطيسي في المستوي الشاقولي دون أن يؤثر هذا التغيير على مواصفات الموجة لأنه عبارة عن اصطلاح. إذا حافظ الحقل الكهربائي على اهتزازه ضمن المحور Y خلال الإنتشار نحصل على مايسمى بالاستقطاب الخطي، إلا أن اتجاه الموجة الكهرطيسية قد لا يكون ثابتاً مع الزمن وإنما متغير فنحصل على ما يسمى استقطاب دائري أو قطع ناقص. يوضح الشكل (1−2) الاستقطاب الدائري لموجة كهرطيسية.



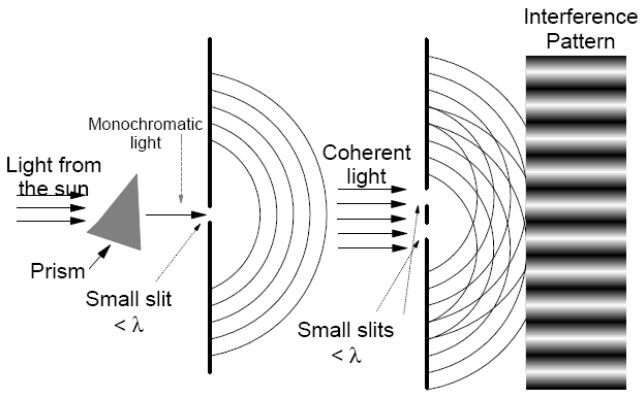
الشكل(1−2). الاستقطاب الدائري لموجة كهرطيسية

نحصل على استقطاب دائري إذا دار الحقلين بزاوية 360ْ عندما تقطع الموجة الكهرطيسية مسافة طول موجة واحد، أما استقطاب القطع الناقص فيحصل إذا كان الدور لا يساوي طول الموجة.

التداخل Interference :

يعتبر التداخل من أهم الظواهر التي تؤخذ بعين الاعتبار عند استخدام الألياف الضوئية في الإتصالات.

- تجربة يونغ:

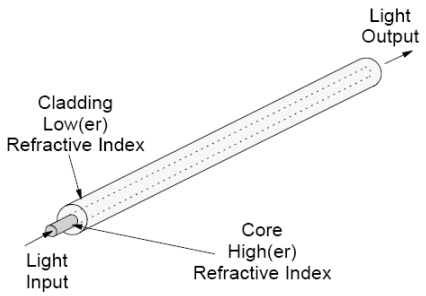


الشكل(1-3). تجربة يونغ

تعتبر من التجارب المهمة تاريخياً وقد قام بها توماس يونغ عام 1802. تنفذ التجربة كما يلي: يتم أولاً تمرير ضوء الشمس عبر موشور للحصول على ضوء وحيد طول الموجة ومن ثم يمرر الضوء وحيد طول الموجة عبر شقين ضيقين كما موضح في الشكل (1-3). يتم إسقاط الضوء الصادر عن الشقين على شاشة والنتيجة التي نحصل عليها هي خطوط متوازية مضيئة ومعتمة.

1.2. إرسال الضوء عبر الليف الضوئي:

الليف الضوئي عبارة عن اسطوانة رقيقة جداً وطويلة من السيلكون، يوضح الشكل (1-4) بنية الليف الضوئي.



الشكل(1–4). بنية الليف الضوئي

نلاحظ من الشكل (1–4) أن الليف يتألف من قسمين أساسين هما: اللب والقشرة. اللب (core) عبارة عن اسطوانة من الزجاج تحيط بها القشرة، ويكون اللب بقرينة إنكسار أعلى من قرينة إنكسار القشرة، وبالتالي فإن السطح الفاصل بينهما يلعب دور مرآة مثالية تحافظ على بقاء الضوء ضمن الليف. عند إرسال نبضة ضوئية عبر الليف فإنها تصل إلى نهايته بمستوى أقل من مستوى الإرسال وتكون أطول في الفترة الزمنية كما أنها تعاني من وجود تشويه عليها، هذه الظواهر ناتجة عمايلي:

- التخامد Attenuation

عند عبور الضوء ضمن الليف فإنه يعاني من تخامد ناتج عن امتصاص الزجاج لقسم من الإستطاعة الضوئية، كما يؤدي وجود الشوائب إلى تبعثر الضوء. تجدر الإشارة إلى أن نسبة الإمتصاص والتبعثر ضمن الليف يتعلقان بطول الموجة.

- الإستطاعة العظمى

وهو حد عملي لقيمة الإستطاعة الضوئية الأعظمية التي يستطيع الليف أن يتحملها.

- الاستقطاب Polarization

الألياف المستخدمة في الإتصالات تكون إسطوانية لكن توجد فيها بعض الشوائب، بالتالي الضوء الذي يمر عبر الليف يتغير استقطابه، في الوقت الحالي تغير الاستقطاب لا يؤثر على عمل الأنظمة الموجودة لكنه سيكون من المشاكل الهامة في الأنظمة المستقبلية.

- التشتت Dispersion

يحدث التشتت عند زيادة عرض النبضة الضوئية وتأثرها بالنبضة التي سبقتها مما يؤدي إلى ظهور تداخل يجعل من عملية استرجاع البتات صعباً . توجد عدة أنواع للتشتت :

- ❖ تشتت المادة: تصدر الثنائيات الضوئية والليزرات حزمة من أطوال الموجة وليس طول موجة وحيد، بالتالي فإن كل طول موجة من النبضة الضوئية له سرعة انتشار مختلفة عن سرعة انتشار طول موجة آخر مما يجعل بعض أطوال الموجة تصل إلى نهاية الليف قبل أطوال موجة أخرى، مما يزيد عرض النبضة الضوئية.
- ❖ تشتت النمط: عند استخدام ليف متعدد الأنماط فإن الضوء يمكن أن يسلك عدة مسارات، ولكل مسار طول مختلف عن الآخر، بالتالي عند إرسال نبضة ضوئية عبر الليف الضوئي فإن كل جزء منها يأخذ مسار مما يجعل هذه الأجزاء تقطع الليف بأزمنة مختلفة مما يزيد من عرض النبضة.
- ❖ تشتت التوجيه: وهو أثر معقد يرتبط ببنية الليف وشكله.

- الضجيج Noise

من المزايا الهامة لليف الضوئي أنه لا يتأثر بمصادر الضجيج الخارجية لأنها لا تدخل إلى الليف. لكن هناك مصادر أخرى للضجيج تأتي من الليف نفسه.

يمكن تحسين أداء الليف بتنفيذ الخطوات التالية:

- ❖ جعل الليف ضيق بشكل كافي بحيث يأخذ الضوء مسار وحيد.
- ❖ اختيار طول الموجة بعناية والأخذ بعين الاعتبار أن التخامد مرتبط بطول الموجة.
- ❖ تقليل عرض طيف الضوء الذي يدخل إلى الليف يؤدي إلى تقليل التشتت.

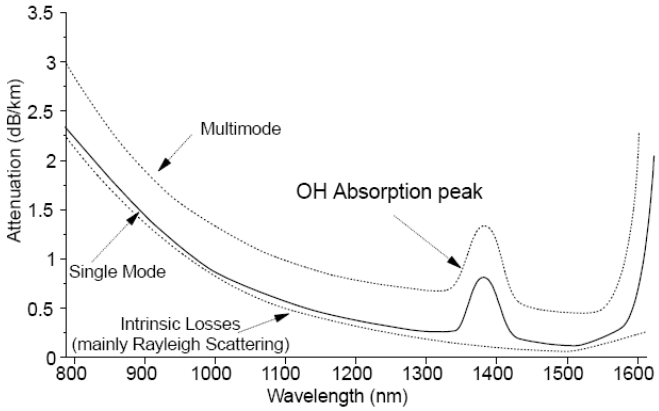
عند استخدام الليف في نظام ما فإننا نهتم بالمزايا التالية:

- ❖ الليف بحد ذاته من حيث ثخانته، قرينة انكساره، ....
- ❖ طول موجة الضوء المستخدم.
- ❖ خصائص مصدر الضوء سواء كان ليزر أو ثنائي ضوئي، وأكثر ما يهمنا هنا هو عرض الطيف الذي يولده هذا المصدر.
- ❖ نوع الكواشف المستخدمة في نهاية الليف.

1.2.1. خصائص الزجاج:

- التخامد:

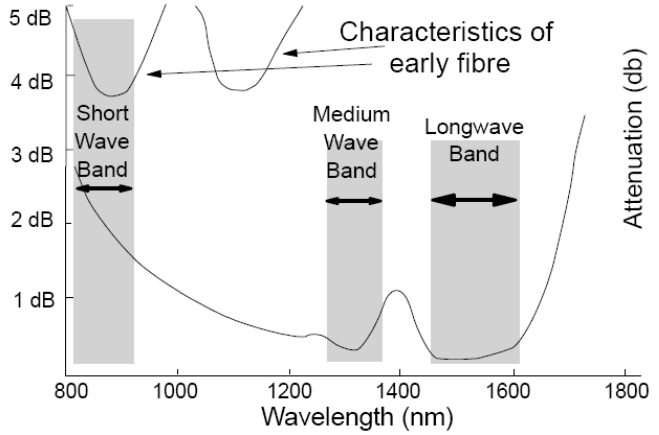
يتعلق الإمتصاص بالتركيب الكيميائي للزجاج المستخدم ضمن الليف، يوضح الشكل (1-5) مخطط لامتصاص الضوء وفق طول الموجة.



الشكل(1-5).مخطط امتصاص الضوء تبعاً لطول الموجة

التخامد الناتج عن التبعثر بمقدار دقيقة (الدقيقة هي عشر طول الموجة)، يسمي بتبعثر Rayleigh. يوجد نوع آخر من التبعثر يسمى بتبعثر Mie وهو ناتج عن عيوب التصنيع من رتبة طول الموجة.

- نافذة الإرسال عبر الليف:



الشكل(1-6).النوافذ المستخدمة لأطوال الموجة

يوضح الشكل (1-6) منحنيات الضياعات بدلالة طول الموجة للألياف الضوئية التي كانت مستخدمة في فترة السبعينيات (المنحني العلوي) والمستخدمة في وقتنا الحالي (المنحني السفلي). بشكل عام توجد ثلاث نوافذ:

❖ نافذة الأمواج القصيرة Shortwave :

وهي الأطوال الموجية الواقعة بين 800-900 نانومتر، وهي من النوافذ المهمة لرخص ثمن المرسلات والمستقبلات فيها. كانت مستخدمة في فترة السبعينيات وبداية الثمانينيات.

❖ نافذة الأمواج المتوسطة Medium Wave :

وهي الأمواج ذات الطول بجوار 1300 نانومتر، تعتبر من النوافذ المهمة في وقتنا الحالي بسبب انعدام التبعثر فيها، المرسلات والمستقبلات أغلى من تلك في نافذة الأمواج القصيرة ولكن قيمة التخامد بحدود 0.4 dB/Km لذلك تستخدم هذه الحزمة في الإتصالات الضوئية بعيدة المسافة.

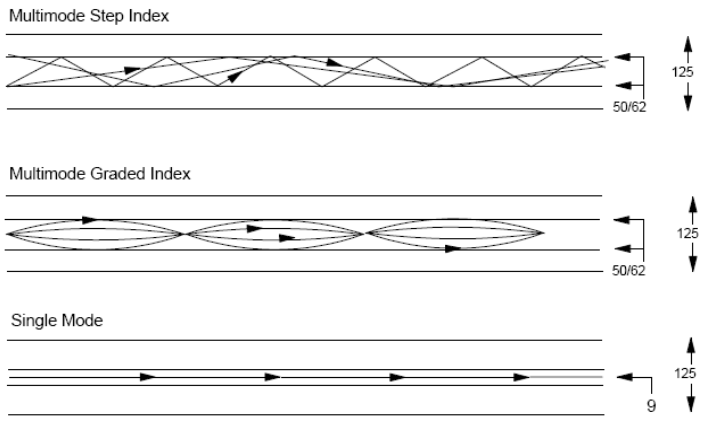
❖ نافذة الأمواج الطويلة Long Wave :

أطوال الموجة في هذه الحزمة تقع بين 1510-1600 نانومتر، تمتاز هذه النافذة بنسبة تخامد قليلة حوالي 0.26 dB/Km، بالإضافة لذلك توجد مضخمات ضوئية تعمل ضمن مجالها لكن من الصعوبة بناء مرسلات ومستقبلات متوافقة معها مما يجعلها غالية الثمن كما أنها تؤدي لتبعثر البقعة الضوئية بشكل كبير . الأنظمة الضوئية الحديثة تعمل ضمن هذه النافذة.

- سعة الارسال:

تمتاز الألياف بسعة إرسال كبيرة، هذه السعة ناتجة عن عرض النوافذ الموضحة في الشكل السابق، حيث تكون نافذة الأمواج المتوسطة بعرض 100 نانومتر وبتخامد 0.4 dB/Km في حين نافذة الأمواج الطويلة بعرض 150 نانومتر وبتخامد حوالي 0.2 dB/Km. ترتبط سعة الإرسال بالتعديل المستخدم (كل نوع تعديل له عرض طيف)، بافتراض وجود تعديل بمعدل 1 Bit/Hz فإن معدل الإرسال يصل إلى $3X10^{13}$ بت بالثانية، لكن نظم الإتصالات الموجودة حالياً تعمل بمعدل نقل لا يتجاوز 10 GBps.

1.3. مبدأ عمل الليف الضوئي :[9]



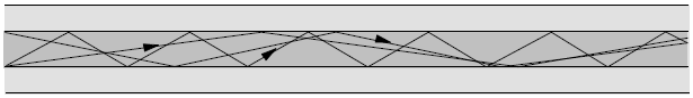
الشكل(1-7) أنماط العمل الموجودة في الألياف الضوئية

يوضح الشكل (1-7) أنماط العمل الموجودة في الألياف الضوئية وهي:

- متعدد النمط بقفزة قرينة الإنكسار Multi-mode .
- متعدد النمط متدرج بقرينة الإنكسار Multi-mode- progressive .
- وحيد النمط Single-mode .

الفرق بين أنماط العمل السابقة هو طريقة انتقال الضوء ضمن الليف. نعبر عادةً عن الليف بقطر كل من اللب والقشرة فمثلاً الليف الأول هو 50/125.

- الليف متعدد الأنماط بقفزة قرينة الإنكسار:

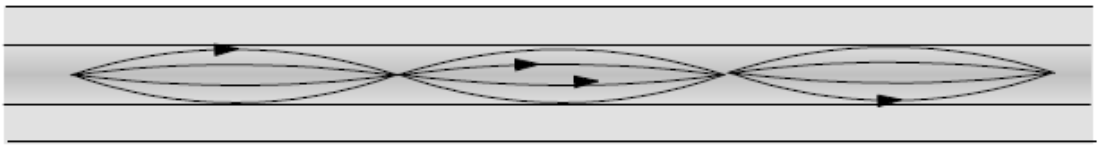


الشكل(1-8). ليف متعدد الأنماط بقفزة قرينة الإنكسار

عدد الطرق الموجودة ضمن الليف محدود، ونسمي كل طريق بنمط. المشكلة في هذا النوع من الألياف أن طول كل طريق مختلف عن الآخر مما يعني أن الضوء سيصل إلى */نهاية الليف بفوارق زمنية تؤدي لزيادة عرض النبضة كما هو موضح بالشكل (1-8).

- الليف متعدد الأنماط بتدرج قرينة الإنكسار:

للتغلب على مشكلة وصول الضوء إلى نهاية الليف بأزمنة مختلفة نجعل قرينة الإنكسار تتغير في اللب بشكل تدريجي ويصبح شكل الطريق كما هو موضح بالشكل (1–9):

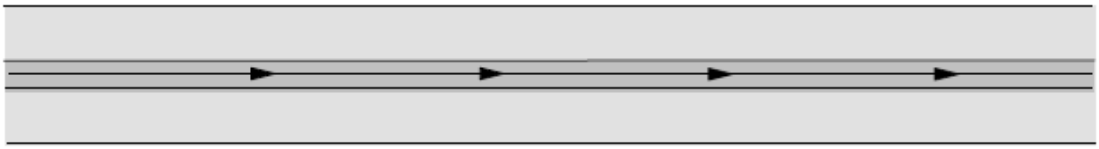


الشكل(1–9). ليف متعدد الأنماط بتدرج قرينة الإنكسار:

تقل قرينة الإنكسار من وسط اللب باتجاه القشرة مما يجعل الضوء الذي ينتقل في الطريق الأقصر ينتشر بسرعة أقل. الهدف من جعل قرينة الإنكسار متدرجة هو جعل النبضة التي تنتشر بأنماط مختلفة تصل جميعها بنفس الوقت.

- الليف وحيد النمط:

هو ليف بقطر صغير للب كما يوضح الشكل (1–10):



الشكل(1–10). الليف وحيد النمط

عند تصغير قطر الليف بشكل كاف فإنه لا توجد انعكاسات مما يجعل الموجة الكهرطيسية تنتشر بطريق وحيد.

1.3.1. انتشار الضوء ضمن الليف :

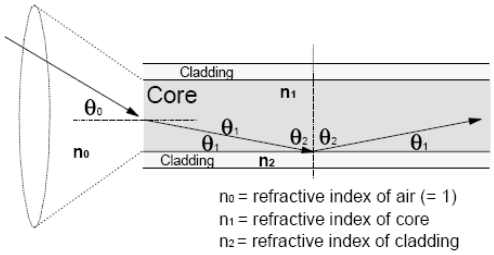
تأخذ مناطق الإنحناءات في الليف بعين الإعتبار عند دراسة الإنتشار ، وغالباً لا يقل نصف قطر الإنحناء عن 2 سم. ينتشر الضوء ضمن الليف دون ضياعات بسبب هذه الإنحناءات، لأن الضوء سينعكس عن السطح الفاصل بين اللب والقشرة في هذه المناطق.

- الزاوية الحرجة Critical angle :

هي الزاوية التي إذا ورد فيها الشعاع الضوئي على السطح الفاصل بين مادتين بقرينتي انكسار مختلفتين فإن هذا الشعاع سينعكس بشكل كلي. تعطى الزاوية الحرجة بالعلاقة التالية:

$$\sin\theta_1 = n_2/n_1$$

- الفتحة العددية NA:



هي الزاوية التي إذا دخل بها الضوء إلى الليف يبقى ضمنه وينتشر إلى نهايته. تعطى بالعلاقة:

$$NA = \sqrt{n_1^2 - n_2^2}$$

تعتبر الفتحة العددية من المزايا الهامة لليف وذلك لأسباب التالية:

- ❖ تعبر عن قدرة الليف على تجميع الضوء ونقله.
- ❖ كلما زادت الفتحة العددية زاد عدد الأنماط الموجودة في الليف.

- الانتشار في ليف متعدد الأنماط:

عند انتشار الضوء في ليف متعدد الأنماط فإنه يسلك عدد محدود من الطرق. يمكننا تفسير ذلك باستخدام معادلات ماكسويل واعتبار الضوء كموجة كهرطيسية. عند انتشار الموجة يجب أن تبقى بطور ثابت على كامل الطريق وبالتالي فإن الموجة تحافظ على الطور اعتباراً من دخولها لليف أي أن المسافة الفاصلة بين نقاط الإنعكاس هي عدد صحيح من طول الموجة، نتيجة لوجود هذا الشرط فإن عدد الطرق الممكنة يكون محدود.

- الإنتشار في الليف وحيد النمط:

من أبسط الشروح الموجود لانتشار الضوء في هذا النوع من الألياف هو القول بأن قطر اللب من مرتبة طول الموجة المستخدم بحيث لا توجد انعكاسات.

الفصل الثاني : أمن الشبكات NETWORK SECURITY

نناقش في هذا الفصل موضوع أمن الشبكات بكافة أنواعها وكيفية اختراقها.

1.1. الشبكات اللاسلكية Wireless Network :[14]

في الشبكات اللاسلكية يوجد لدينا مرسل / مستقبل راديوي ونميز في هذه الشبكات بين نوعين من الاختراق:

- اختراق سلبي Passive

يكون الهدف في هذا النوع من الاختراق هو التنصت على الشبكة، بمعنى آخر تكون الغاية هي سرقة البيانات المتبادلة بين الطرفيات، فبعد معرفة مواصفات المرسل من حيث التعديل والتشفير، يتم استخدام مستقبل راديوي مناسب ليبدأ باستقبال البيانات.

يتميز الاختراق السلبي بأنه غير قابل للكشف حيث لا يؤدي دخول الشبكة إلى التغيير من حالة الاشارات المتبادلة بين المرسل/المستقبل.

تتم الحماية في هذا النوع من الاختراق بتطبيق تقنيات التشفير، وكذلك باستخدام تقنيات الطيف المنثور (القفاز الترددي Frequency Hopping، النثر بالسلاسل DSSS)، نذكر مثال عن هذا النوع من الاختراق ما قامت به شركة غوغل من سرقة لبيانات مستخدمي انترنت لاسلكي في عام 2010.

-اختراق ايجابي Active

يحدث في هذا النوع من الاختراق مداخلة على الشبكة، أي يتم استخدام أحد الخدمات التي تقدمها الشبكة بشكل غير مشروع. يتطلب هذا النوع من الاختراق المعرفة المسبقة بميثاق الشبكة Protocol كما أنه سهل الكشف.

1.2. الشبكات السلكية Wired Network:

يتم الاختراق في الشبكات السلكية بطريقتين أساسيتين تتفرع عنهما طرق عديدة كما سنرى في شبكة الألياف الضوئية :

- وصلة مباشرة على الأسلاك

تتميز هذه الطريقة بسهولة كشفها حيث تتغير مستويات الاشارة المستقبلة عند كل طرفية من طرفيات الشبكة.

- استقبال الأمواج الكهرطيسية الصادرة عن الأسلاك

تتميز هذه الطريقة بصعوبة كشفها حيث أن المخترق في هذه الحالة غالباً ما يريد سرقة البيانات المتبادلة بين طرفيات الشبكة، ويتم الاستفادة من خصائص مرور إشارة كهربائية في ناقل حيث تؤدي لتوليد موجة كهرطيسية معروفة المواصفات (باستخدام معادلات ماكسويل).

1.3. شبكات الألياف الضوئية Fiber optic Network :[16],[10],[5]

شبكة الألياف الضوئية هي شبكة سلكية بالأساس، وبالتالي فإن طرق الاختراق في الشبكات السلكية تبقى صالحة مع مراعاة وسط النقل.

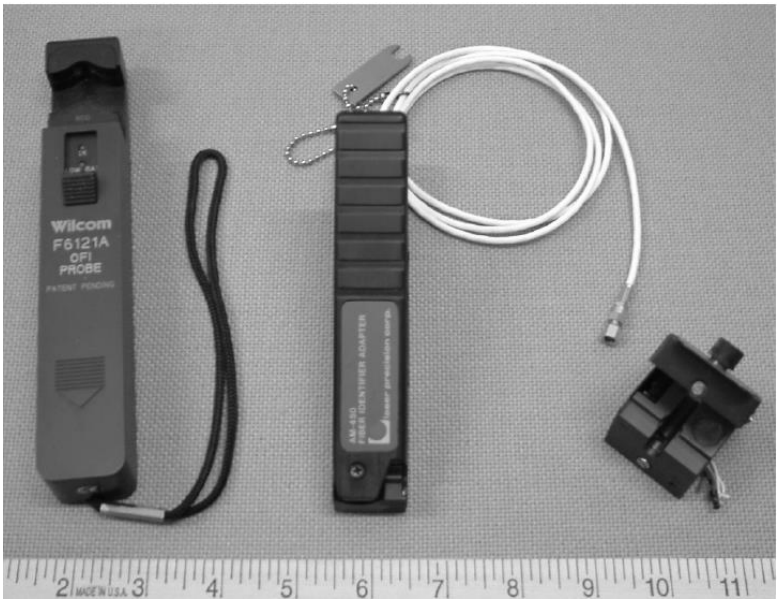
وموضوعنا الأساسي في هذا البحث هو بناء نظام لكشف هذه الاختراقات الحاصلة في الشبكات الضوئية.

1.1.3 . وصلة مباشرة:

تحدث هذه العملية عن طريق الاختراق الفيزيائي المباشر لليف. حيث تقوم عملية الاختراق بتركيب وصلة مباشرة على الليف بشكل حرف T، مهمة هذه الوصلة هي توزيع الإشارة الضوئية الواردة إلى إشارتين متماثلتين شكلاً وباستطاعة أقل من الإشارة الأصلية، تتجه إحداهما إلى المستقبل الأصلي، والأخرى إلى المخترِق.

تتم عملية الاختراق بطريقتين :

- عن طريق تقشير لحاء الليف ومن ثم توصيل ليف جديد. تتميز هذه الطريقة بأنها تتطلب دقة عالية في التنفيذ لكن أدوات التنفيذ رخيصة ومتوافرة بكثرة، يبين الشكل (2-1) مجموعة من الأدوات المستخدمة في تنفيذ هذا النوع من الاختراق.
- عن طريق مجموعة من المغارز، موصولة بمجمع يتبعه ليف ضوئي من جهة، ومن الجهة الأخرى تغرز في الليف المقصود لتجمع جزءاً من الإشارة الضوئية لترسل إلى الجهة المخترقة.



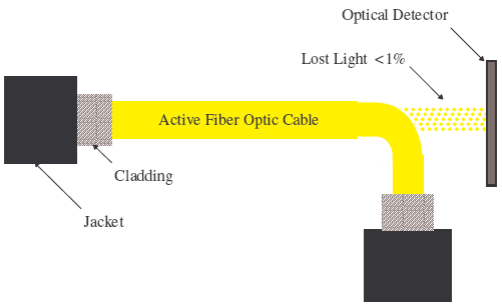
الشكل(2-1).أدوات الاختراق لتنفيذ وصلة مباشرة

تؤدي عملية الاختراق بهذه الطريقة إلى تناقص في قيمة الاستطاعة في المستقبلة في نهاية الليف، نتيجة نفاذ قسم منها إلى جهة الاختراق. ويحدث ايضاً تغير في توزع الاستطاعة الضوئية في نهاية الليف كما يُضاف ضجيج إلى الاستطاعة الضوئية المستقبلة ناتج عن تغير في قيمة قرينة الانكسار في اللب في منطقة الاختراق. يؤدي هذا الضجيج بدوره إلى زيادة معدل الخطأ في نظام اتصالات الألياف الضوئية.

ولمقاومة هذا النوع من الاختراقات فإنه يمكن مراقبة كل من مستوى الاستطاعة الضوئية المستقبلة وتوزعها مكانياً وكذلك مراقبة معدل الخطأ في نظام الاتصالات الضوئي.

2.1.3. الاختراق بثني الليف:

تتم عملية الاختراق في هذه الطريقة بثني الليف بنصف قطر محدد (بحيث يبقى ضمن مجال مرونة الليف)، يؤدي ثني الليف إلى تفاذ جزء من الأشعة الضوئية المارة في اللب إلى الغلاف ومنها إلى الوسط الخارجي، تقدر قيمة الاستطاعة الخارجة من الليف بحوالي 1 dB، يتم تركيب مستقبل ضوئي يحول الاشعة الضوئية الصادرة عن الليف إلى معلومات ذات معنى، يبين الشكل (2-2) رسماً توضيحيا لعملية الاختراق بهذه الطريقة.



الشكل(2-2).عملية الاختراق بواسطة ثني الليف

تتميز هذه الطريقة بصعوبة تنفيذها وحاجتها إلى حرفية عالية ، كما أن نجاح عملية الاختراق يتطلب معرفة مسبقة ببنية نظام الاتصالات كاملة (تراميز كشف الخطأ، التعديل، ...). بما أن تناقص الاستطاعة هو مايميزهذا النوع من الاختراق بالتالي مراقبة الاستطاعة الضوئية في طرف الاستقبال تفيد في عملية الكشف. لكن ثني الليف لا يؤدي فقط إلى تناقص قيمة الاستطاعة المستقبلة (كقيمة متوسطة) وإنما إلى تغير التوزع المكاني للكثافة الضوئية المستقبلة، بالتالي مراقبة توزع الاستطاعة المستقبلة تساهم أيضاً في تعزيز الحماية ضد هذا النوع من الاختراقات. مما سبق نجد أنه للحماية من هذا النوع من الاختراق فإن على نظام الحماية مراقبة كل من الاستطاعة الضوئية المستقبلة ومراقبة توزعها مكانياً.

3.1.3. الطريقة الكمومية :

تستخدم هذه الطريقة في الاختراق لواقط خاصة تتوضع حول الليف في منطقة الاختراق.

- يتم اختراق الضوء داخل الليف بوساطة لاقط خاص متوضع حوله دون الحاجة إلى تعديل في الليف نفسه. بالاستفادة من الخواص الكمومية للمادة، يستطيع هذا اللاقط تجميع بعضاً من الإشارة الضوئية المتسربة عبر الليف والاستفادة منها في الاختراق.

- كشف هذه الطريقة يكون بدراسة التحليل الجزيئي للمادة (الكوانتم) وبالتالي تحديد أماكن تسرب الضوء من داخل الليف.

1.4. حماية الشبكات من الاختراق:

في الشبكات السلكية واللاسلكية تتم الحماية على مستوى البيانات وذلك باتباع الطرق التالية:

- تشفير المعطيات Cipher: خوارزمية تشفير صعبة الكسر.
- تغيير مفاتيح التشفير كل فترة زمنية.
- استخدام تقنيات الطيف المنثور (مع مولدات شبه عشوائية ذات دور كبير جدا).
- استخدام التحجيب في الشبكات السلكية Shield حيث تضعف استطاعة الأمواج الكهرطيسية الناتجة عن الاسلاك.
- مراقبة مستويات الاشارة.
- استخدام ميثاق Protocol خاص بتبادل المعطيات يتطلب ارسال واستقبال من كل طرفية من طرفيات الشبكة.

الفصل الثالث : المبادئ العامة في القياسات الضوئية

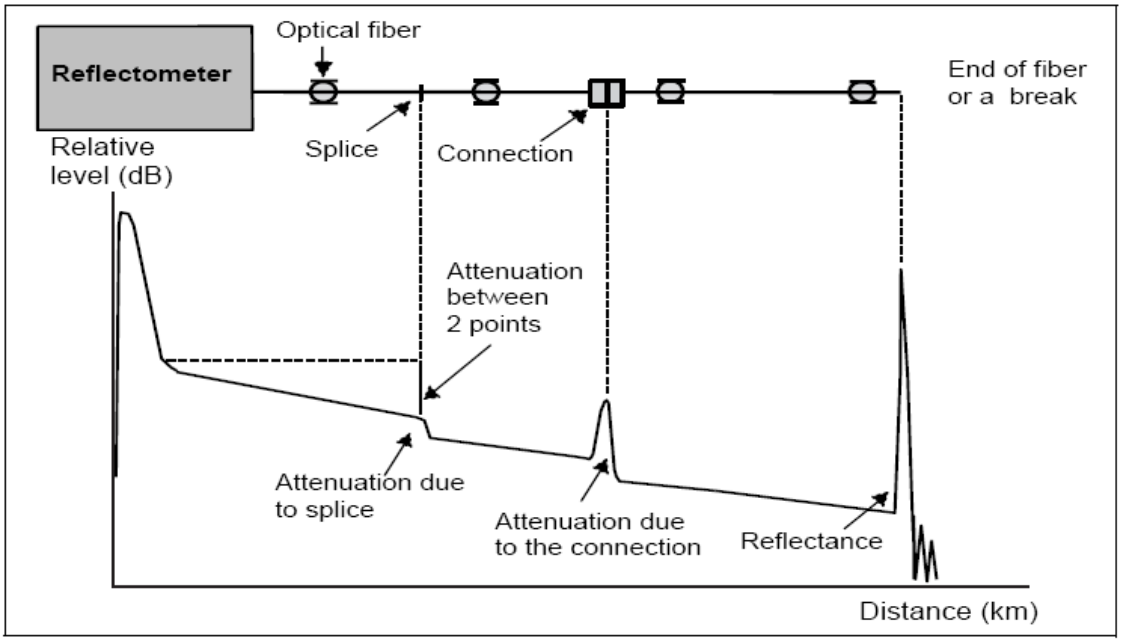
سندرس في هذا الفصل المبادئ الأساسية المستخدمة في عمليات القياس ، لذلك سنتحدث عن القياسات المأخوذة من جهاز قياس الانعكاس reflectometer (OTDR) ، ومحلل الطيف spectrum analyzer (OSA) ، ومحلل قياس تشتت القطبية PMD .[4]

وتشمل هذه الدراسة المبادئ التالية :

- مبادئ القياس باستخدام جهاز قياس الانعكاس.
- مبادئ القياس ثنائي الاتجاه.
- مبادئ تحليل الطيف.
- مبادئ قياس تشتت القطبية.
- مبادئ قياس التشتت اللوني.
- مبادئ قياس الاستطاعة الضوئية والضياعات.
- معايير أنواع الألياف الضوئية.

3.1. القياس باستخدام جهاز الانعكاس:

يقوم مبدأ هذا الجهاز بحقن نبضة ضوئية في نهاية الليف من أجل الدراسة والتحليل. وفي الوقت نفسه تنتشر الكثافة الضوئية في الاتجاه المعاكس لانتشار النبضة. ويكون شكل الإشارة الناتجة أسي متناقص كما في ظاهرة التبعثر المرتد، مع بعض القمم نتيجة الانعكاسات الناتجة



الشكل(3-1).جهاز الانعكاس

عن الوصلات داخل الليف.

ونتيجة لهذا القياس فإننا نحصل على المعلومات التالية :

- مقدار الضياعات.
- موقع المشكلة وبعدها عن بداية الليف.
- نسبة الضياعات dB/Km.
- مقدار الانعكاسات على الخط.

وبالتالي تحديد مكان المشكلة أو الاختراق داخل الليف الضوئي.

- وبالانتقال إلى مدى صحة هذا القياس فإن UTI-T قد اعتبرت هذا القياس مبدأً لقياس الضياعات ضمن الليف الضوئي ، ولكن ضمن شروط معينة يجب الأخذ بها وهي :
  i. عند الحقن: يجب أن تكون انعكاسات فرينل محدودة عند مدخل الليف.
  ii. يجب استخدام منبع ضوئي عالي الاستطاعة (منبع ليزري).
  iii. عرض مجال الاستقبال يجب أن يحقق الأمثلة بين استطاعة النبضة واستطاعة الضجيج.
  iv. الاستطاعة المنعكسة يجب تمثيلها لوغاريتمياً.

الانعكاس :

يعرف عامل النعكاس على أنه نسبة الاستطاعة المنعكسة إلى الاستطاعة الكلية الواردة. أما أسبابه الرئيسية فهي الانكسارات المتتالية للضوء على طول الليف المستخدم في الاتصال، وإذا لم تؤخذ بعين الاعتبار فإن أداء النظام سينخفض بشكل ملحوظ. حيث يمكن أن تسبب كثرة الانكسارات غير المسيطر عليها بزيادة نسبة الضجيج في المستقبل مما يؤدي لانخفاض في أداء النظام.

ولحساب قيمة معامل الانعكاس يجب قياس المطال الكلي لانعكاس فرينل، واستخدام صيغ التحويل للحصول على قيمة الاستطاعة المنعكسة. حيث تأخذ هذه الصيغة بالاعتبار:

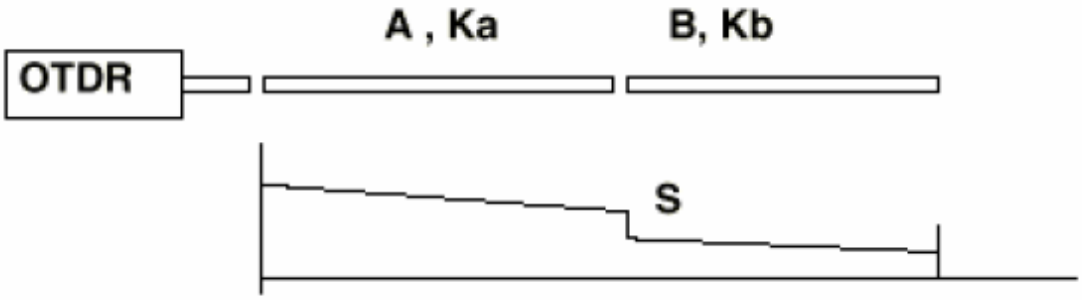
- المطال الكلي المنعكس المقاس بمقياس الانعكاس.

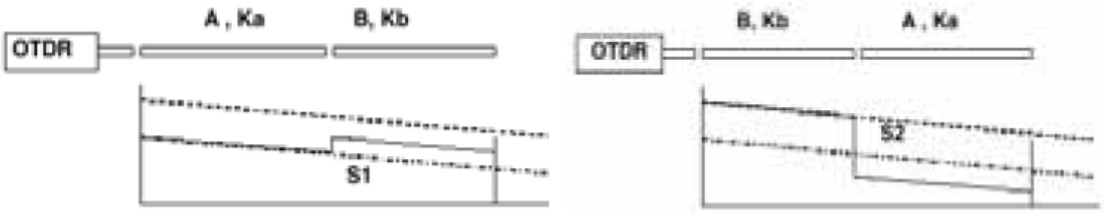
- عرض النبضة المستخدمة في قياس الانعكاس.
- معامل التبعثر لليف الضوئي المستخدم :

- معامل التبعثر بالنسبة لنبضة 1 ns وليف أحادي النمط 1310 nm to 79 dB-
- 1550 and 1625 nm to 81 dB-
- بالنسبة لليف متعدد الأنماط : 850 nm to 70 dB-
- 1300 nm to 75 dB-

3.2. مبادئ القياس ثنائي الاتجاه :

عندما نستخدم أنماطاً مختلفة من الألياف الضوئية بمعاملات مختلفة فإن جهاز قياس الانعكاس سيعطي قيمة كبيرة للتبعثر عند وصلها معاً، وتتبع هذه القيمة قيم معاملات التبعثر لكل ليف على حدى.





$Ka < Kb\ Kb-Ka=\Delta K\ S1=S+\Delta K$ $Ka < Kb\ Kb-Ka=\Delta K\ S2=S-\Delta K$

الشكل(3-2).الضياعات في الأنماط المختلفة لليف

وبالتالي لحساب قيمة معامل الضياع في هذه الحالة فإنه متوسط قيم المعاملات لكل ليف :

$S=(S1+S2)/2$.

حيث نقوم بحساب معاملات الليف الأول لوحده، ثم نقوم بحساب معاملات اليف الثاني لوحده، بعد ذلك نقوم بحساب المتوسط لهما.

3.3. مبادئ تحليل الطيف :

تقنية التقسيم الموجي WDM(wavelength division multiplex) هي تقنية فعالة جداً في الاتصالات الضوئية، لأنها لاتحتاج لتعريف خطوط نقل جديدة ولا تهتم بسرعة النقل. حيث تنتقل المعلومات في هذه الحالة بأطوال موجات مختلفة بقنوات مختلفة، وقد تم تعريف هذه القنوات من قبل ITU-T بالتوصيات G-692 .

وتهتم هذه التقنية عند القياس بالمعاملات التالية:

- يجب التطابق بين القناة وطول الموجة الموافق دون حصول انزياحات.
- القيمة الصحيحة لاستطاعة القناة دون ضياعها.
- الحصول على قيمة مقبولة لنسبة الإشارة إلى الضجيج. SNR

وللحصول على هذه المعاملات فإننا نستخدم محلل الطيف الضوئي OSA(optical spectrum analyzer).

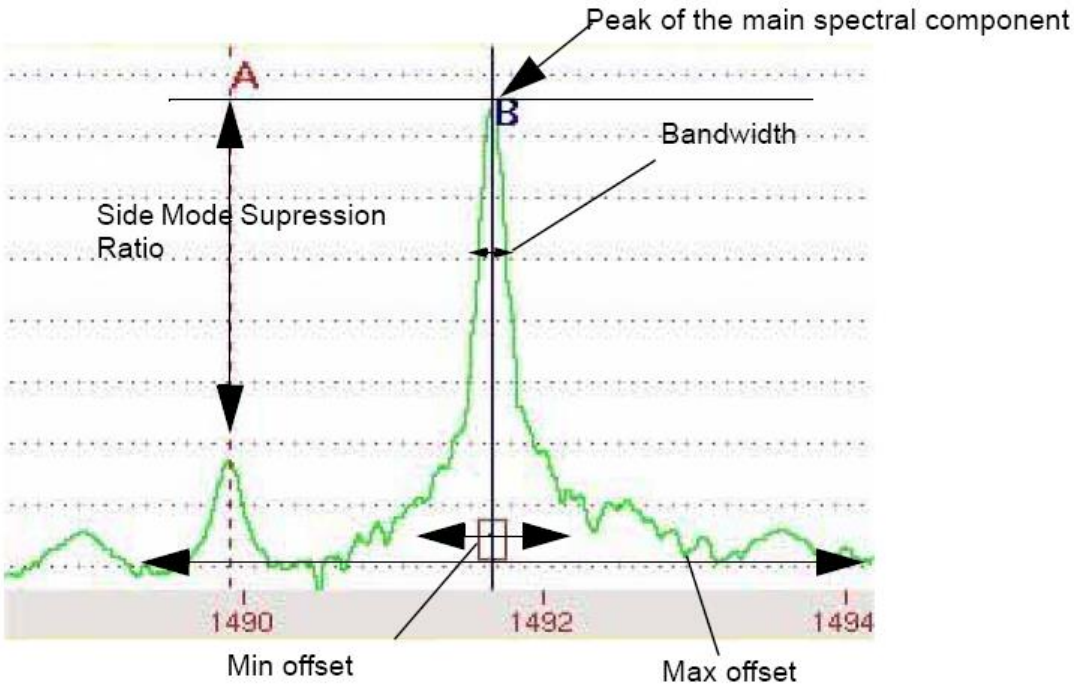
قياس النتائج :

يقوم محلل الطيف الضوئي بعرض الطيف كاملاً لكل القنوات، وعرض جدول بقيم كافة المعاملات لكل قناة. حيث يقوم بكشف وقياس اتوماتيكي لكل قناة ويعرض عدد هذه القنوات واستطاعة كل منها بالإضافة لـ :

- طول الموجة.
- مقدار التداخل بين القنوات.
- مستوى الاستطاعة.
- نسبة الإشارة إلى الضجيج.

ويمكن أيضاً حساب الاستطاعة الكلية.

- تحليل DFB : نقوم بتحليل القياسات باستخدام منابع DFB الليزرية من أجل الحصول على أفضل قيمة لنسبة الخطأ المرتكب. وتتضمن هذه القياسات :
  - SMSR: الفرق المطالي بين المركبة الأساسية وأعلى مركبة جانبية.
  - Mode Offset: مقدار الانزياح بين الموجتين.
  - Peak amp: مستوى الاستطاعة الأعلى في المركبة الرئيسية .
  - BW: عرض المجال المستخدم من قبل المرسل الليزري DFB.



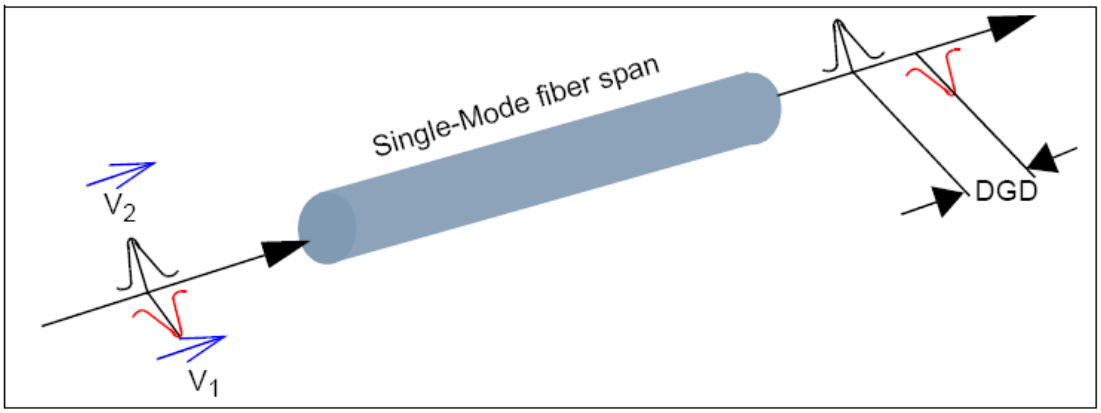
الشكل(3-3).قياسات محلل الطيف

3.4. مبادئ قياس تشتت القطبية:

إن سرعة النقل والمدى هما المعاملان الأساسيان في الاتصالات الضوئية، لذلك يجب الأمثلة بينهما دوماً.

وعندما تزداد القنوات كثيراً وتزداد معها الموجات المختلفة، أو تزداد السرعة فوق 10 Gb/s فإننا بحاجة دراسة تشتت القطبية وتحديد معاملات قياسها.

في حالة الليف وحيد النمط فإن PMD تشتت القطبية يؤثر في مطال معدل النقل، وذلك ناتج عن الاختلاف في أزمنة انتشار طاقة الأمواج المستخدمة. وهذا الاختلاف ناتج عن شكل الليف نفسه والضغط المحيط به.



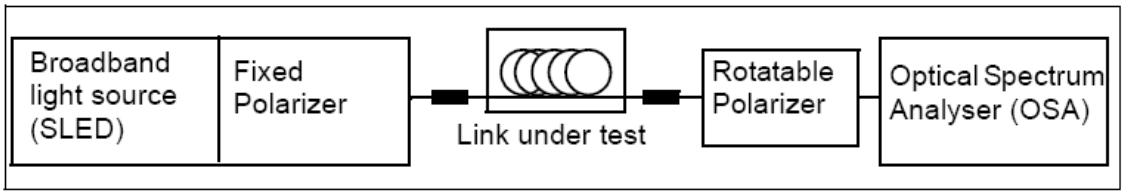
الشكل(3-4).انعكاس القطبية

كما يعبر مقياس تشتت القطبية عن تأخير المجموعة DGD .

وتأخير المجموعة يساهم في زيادة عرض مجال الإرسال، مما يؤدي لزيادة نسبة الخطأ، وبالتالي فإن الإرسال يتحدد بسرعة معينة لايمكن زيادتها في الخط. أي أنه يجب البحث عن القيمة الأمثلية لـ PMD لتحديد سرعة النقل المناسبة.

طريقة قياس PMD :

يتطلب قياس القطبية منبعاً للقطبية عريض المجال في الطرف الأول، ومحلل طيف مستقطب في الطرف الثاني.



الشكل(3-5).طريقة قياس PMD

ومن قياس المطال المعدل من محلل الطيف نستطيع باستخدام تحويل فوريه السريع FFT قياس تشتت القطبية.

طريقة تحويل فورييه السريع تعطي في توزيعها الزمني منحني غاوس ومن هذا المنحني نستطيع قياس DGD تأخير المجموعة.

وجهاز القياس يجب أن يمتلك مجالاً أكبر من الليف نفسه، ومجال 35dB كافية في معظم الأحيان حتى في الشبكات بعيدة المدى.

والجدول التالي يوضح قيم PMD بالنسبة للمدى:

Bit rate (Gbit/ s)	**Maximum PMD (ps)**	**PMD coefficient (ps/√ km)** 400 km cable length
2.5	40	< 2
10	10	< 0.5
40	2.5	< 0.125

3.5. مبادئ قياس التشتت اللوني :

التشتت اللوني ينتج عن الانكسارات التي تصيب الضوء المنتقل عبر الليف، وبالتالي يحدث تأخيرات في الإشارة المرسلة مما يؤدي إلى زيادة عرض نبضة الإرسال وبالتالي زيادة معدل الخطأ BER.

وسرعة الإرسال ترتبط بطول الموجة، وانزياح الأمواج يسبب فراغات غير متساوية بين القنوات، وبأخذ قيمة حوالي 1550 nm تكفي لجعل هذه الفراغات خطية.

ويعرف التشتت اللوني بالمعاملات التالية :

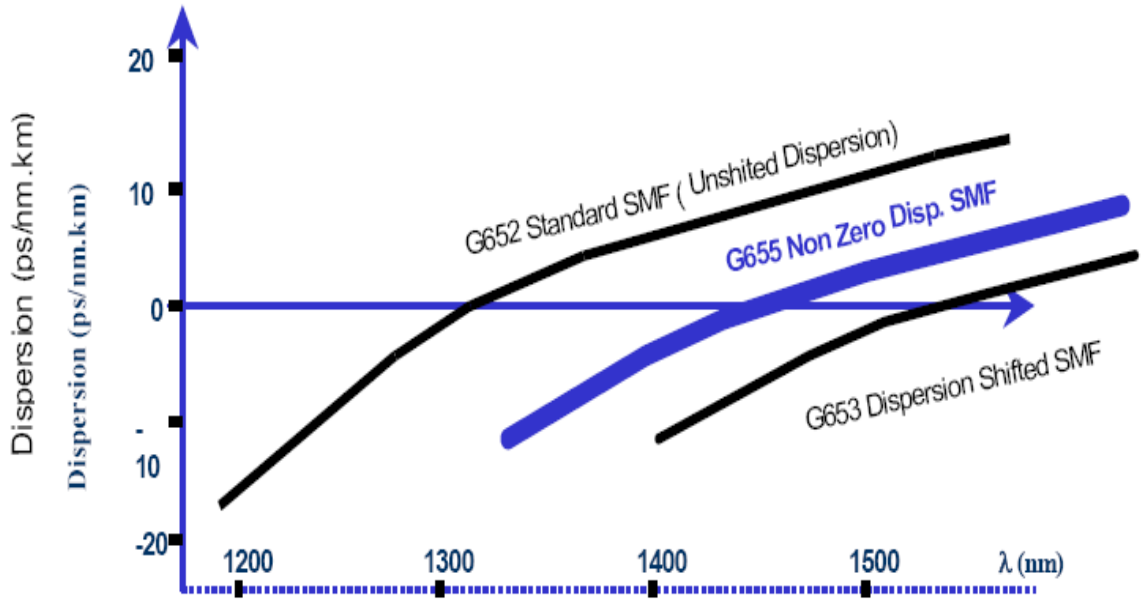
- التأخير عند طول موجة محدد (ps).
- معامل التشتت D يقاس بـ ps/nm ، وهو يتبع التأخير المعرف كتابع لطول الموجة.
- الميل S يعرف بـ $ps/(nm^2.km)$.

ومن المهم أن نعرف علاقة المدى بالدقة لأن خطأً مقداره 10% في طول الليف يقابل 10% خطأ في قياس التشتت.

والتشتت اللوني واحدة من أهم مواصفات الليف، حيث تعتمد بشكل أساسي على طريقة التصنيع.

والجدول التالي يوضح بعض القيم

Type of fibre/Corresponding standard	Coefficient of dispersion at 1550 nm
Single-mode standard/ITU-T G.652	+17 ps/(nm.km)
Single-mode offset dispersion/ITU-T G.653	0 ps/(nm.km)
Single-mode non-zero offset dispersion/ITU-T G.655	+3 ps/(nm.km)



الشكل(3-6).معايير ITU-T

وفي معايير ITU-T تؤخذ المميزات التالية :

- في حالة تشتت صفري يكون طول الموجة بين 1300-1320 nm.
- وتكون القيمة العظمى للميل (nm².km)/ps 0.093-.
- يأخذ معامل التشتت قيمتان صغرتان بين 1260-1360 nm.

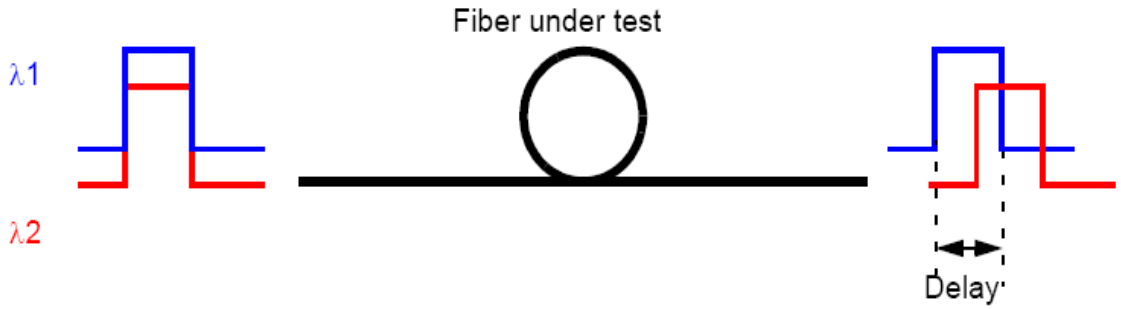
وبالتالي فإن معامل التشتت لايعتمد على الزمن أو نوع الليف، ولا يتأثر حتى بتغيرات درجة الحرارة.

والعامل الوحيد الذي يمكن أن يؤثر على معامل التشتت اللوني هو الإشارة المرسلة نفسها.

وفي محلل التشتت يُعطى تأخير الليف كتابع لطول الموجة، ومن خلال ذلك يمكن حساب معامل التشتت، والميل وطول الموجة الموافق للتشتت الصفري.

ننتقل الآن إلى طرق قياس التشتت اللوني المختلفة ، ونتحدث عن طريقة الوحدة الأساسية:

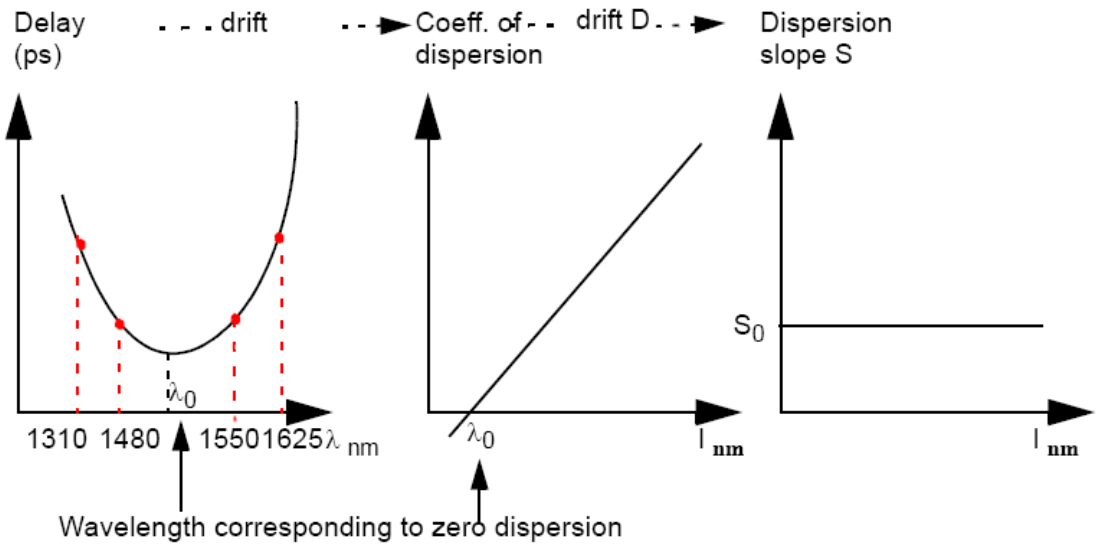
حيث تقوم هذه الوحدة بقياس تأخير انتشار النبضة على طول الليف الناتج عن عوامل الانعكاس نتيجة الوصلات الموجودة ، فنأخذ أربعة أطوال موجية ونستخدم أحدها كمرجع.



الشكل(3-7).تأخير الموجة

وبالاعتماد على النتائج الأربعة نعرف منحني التأخير تبعاً لطول الموجة باستخدام إحدى الخوارزميات :

- $A+B\lambda+C\lambda^2$
- $A+B\lambda 2+C\lambda-2$
- $A+B\lambda 2+C\lambda-2+D\lambda 4+E\lambda-4$



الشكل(3-8).منحني التأخير

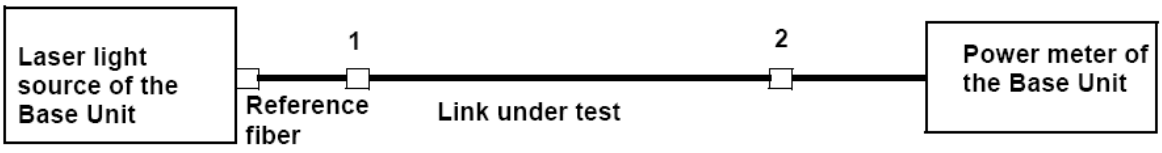
ومن خلال المنحني نستطيع حساب الانزياح وبالتالي حساب معامل التشتت والميل.

3.6. مبادئ قياس الاستطاعة الضوئية والضياعات OFI :

لقياس الاستطاعة المرسلة أو المستقبلة نحتاج فقط إلى مقياس الاستطاعة . حيث نصل المقياس إلى بداية الليف في نهاية المرسل لقياس الاستطاعة المرسلة، ونصله إلى نهاية الليف بداية المستقبل في حالة قياس الاستطاعة المستقبلة.

قياس الضياعات :

يتطلب ذلك مقياس الاستطاعة مع المرسل الضوئي. ونستطيع استنتاج الضياع الحاصل عند قياس الاستطاعة في نقطتين مختلفتين.



$$\text{Attenuation } A_{(dB)} = P1_{(dBm)} - P2_{(dBm)}$$

الشكل(3-9).قياس الاستطاعة

وهذا يتطلب ما يلي:

- استخدام منبع ضوئي مستقر بالنسبة للزمن والحرارة.
- جميع الوصلات يجب أن تكون نظيفة.
- استخدام ليف مرجعي في المنطقة بين المنبع والنقطة الأولى.

- بداية نصل المقياس عند النقطة الأولى ونقيس الاستطاعة P1.

- ثم نصل الليف المراد اختباره وفي نهايته المقياس لقياس P2.
- الفرق بينهما هو الضياع الحاصل نتيجة الليف.

ومن المفيد استخدام وصلات من نوع واحد خلال عملية القياس.

3.7. معايير الألياف الضوئية :

Fibre	non-offset dispersion	offset dispersion	non-zero or homogeneous offset dispersion
Standard ITU/Y	ITU-T G.652	ITU-T G.653	ITU-T G.655
Standard IEC	IEC 60793-1-1 type B1	IEC 60793-1-1 type B2	IEC 60793-1-1 type B3
Standard TIA/EIA	Iva	IVb	IVb
Approximate λ_0 .	1310 nm	1550 nm	1500 nm or indefinite

Fibre	non-offset dispersion	offset dispersion	non-zero or homogeneous offset dispersion
1310 nm zone	Sellmeier 3-term	Sellmeier 5-term	Sellmeier 5-term
1550 nm zone	Sellmeier 5-term	Quadratic	Sellmeier 5-term
Complete curve	Sellmeier 5-term	Sellmeier 5-term	Sellmeier 5-term

الفصل الرابع :الجزء العملي

نبين في هذا الجزء بنية نظام كشف الاختراق المقترحة ومحاكاة تنفيذها بوساطة برنامج الـ MATLAB والنتائج التي حصلنا عليها بتطبيق برنامج اختباري على بيئة NET..

1.1. بنية نظام الحماية:

لتأمين حماية الليف يجب أن يحوي نظام كشف الاختراق على الأنظمة الجزئية التالية:

1. مراقبة الاستطاعة الضوئية:

تعتمد هذه الطريقة على استخدام حساسات لقياس الاستطاعة الضوئية تُركّب عند نهاية الليف الضوئي وفي مركزه تحديدا حيث تتركز الاستطاعة وتتناقص عند الأطراف. يتم تحديد عتبة التنبيه إذا انخفضت قيمة الاستطاعة المستقبلة عن الحد الطبيعي، فما هي هذه القيمة وكيف تحسب؟

لنفترض أنه لدينا نظام اتصالات باستخدام ألياف ضوئية، وأن الليف المستخدم بطول يساوي L كم، وبمعامل تخامد x dB/Km، وأن استطاعة الارسال هي P_t (dB) عندئذ تكون قيمة الاستطاعة المستقبلة P_r تعطى بالعلاقة التالية:

$$P_r = P_t - L*x$$

بهذه الطريقة يتم تحديد قيمة عتبة التنبيه. يمكن تحديد نوع الاختراق إذا كان بثني الليف أو بالاختراق المباشر بتحديد مقدار التناقص فإذا كان بجوار 1 dB قلنا أن الاختراق بالثني وإلا فإنه بالاختراق المباشر.

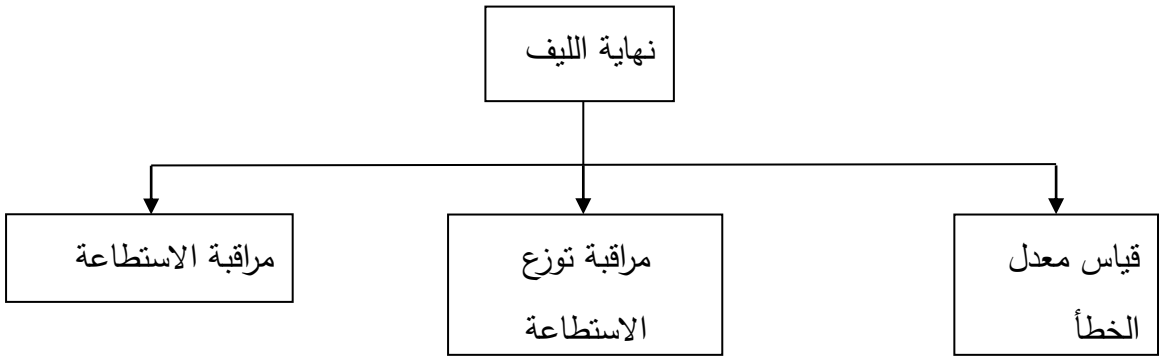
2. مراقبة توزع الاستطاعة الضوئية مكانياً:

تتم عملية المراقبة باستخدام مجموعة من حساسات الاستطاعة، يجب تحديد توزع الاستطاعة الضوئية بشكل مرجعي عند تركيب الليف، بعدها يتم بشكل مستمر مراقبة قيم الاستطاعات الضوئية المستقبلة في الحساسات ومقارنتها مع الحالة المرجعية، يتم اعطاء تنبيه عند التغير في توزع الاستطاعة الضوئية المستقبلة.

3. مراقبة معدل الخطأ:

كما نعلم فإن من محددات أي نظام اتصالات معدل خطأ يقبل بوجوده، يتم تحديد معدل الخطأ حسب نوع المعلومات المنقولة ضمن نظام الاتصالات توفيرها. يتم تقدير معدل الخطأ باستخدام تراميز كشف الخطأ وتصحيحه ويعطي تنبيها عند زيادة معدل الخطأ عن المعدل المطلوب.

يبين الشكل (4-3) المخطط الصندوقي لنظام كشف الاختراق المقترح:



الشكل(4-3).المخطط الصندوقي لنظام كشف الاختراق

1.2. التنفيذ العملي:

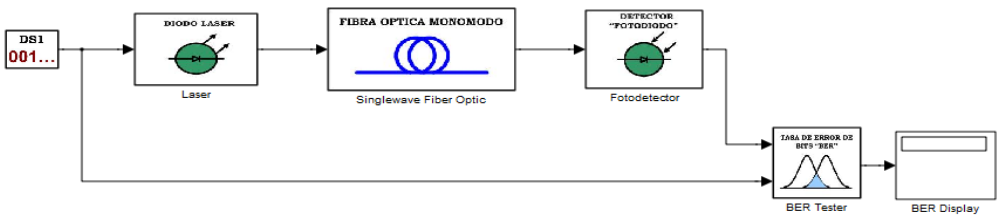
تم تنفيذ محاكاة لبنية النظام المقترحة بوساطة برنامج الـ MATLAB.

- محاكاة توزع الاستطاعة الضوئية مكانياً في نهاية الليف:

تكون الاستطاعة الضوئية أعظمية في مركز الليف وتتناقص تدريجياً باتجاه الحواف وهذا ما تبينه نتائج المحاكاة التي تمت لانتشار الضوء في الليف الضوئي، أما عند حدوث اختراق فإن توزع الاستطاعة يتغير، تبقي القيمى العظمى للاستطاعة متمركزة في نهاية الليف لكن تظهر قيم محلية عظمى باتجاه حواف الليف، إن تغير توزع الاستطاعة ينتج عن تغير نمط الانتشار ضمن الليف سواء نتيجة الثني أو ال واللحام (تغير في كثافة مادة اللب ينتج عنها تغير في قرينة الانكسار)، نحصل على قيمة عظمى في المركز حيث تكون الاشعة الضوئية متمركزة لكن ونتيجة التشوه الحاصل في الليف فإن بعض الأشعة الضوئية لا يمكن لليف توجيهها لتكون في المركز، ينتج عن ذلك ظهور قيم محلية عظمى وكذلك انخفاض في قيمة الاستطاعة الكلية.

- محاكاة نظام اتصالات:

يبين الشكل (4-4) مخطط صندوقي يوضح بنية نظام اتصالات بسيط يعتمد على الالياف الضوئية:

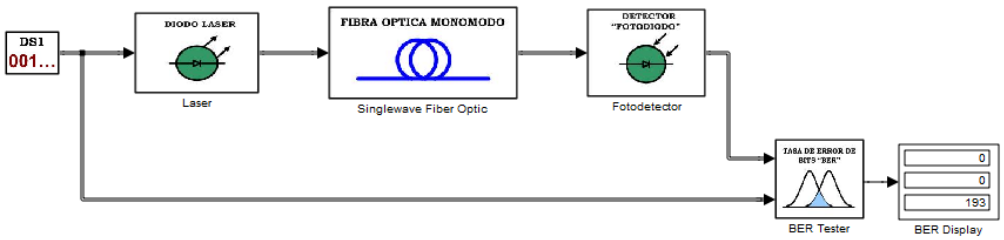


الشكل(4-4). بنية نظام اتصالات ضوئي بسيط

يتألف النظام من:

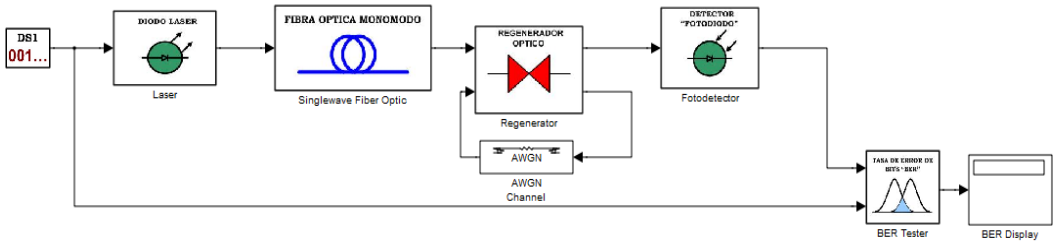
1. مصدر معلومات: منبع معلومات رقمي يمثل المعلومات التي يتعامل معها النظام.
2. مرسل ضوئي: في النظام المبين أعلاه هو عبارة عن ديود ضوئي، مهمة المرسل تحويل الاشارة الرقمية إلى إشارة ضوئية، يتم ذلك باستعمال طرق التعديل الضوئي، التعديل في حالتنا هو OOK (On/Off Keying)، طول الموجة المرسلة 1550 nm.
3. الليف الضوئي: وهو وسط الانتشار ويكون بتخامد منخفض كما أنه يتمتع بعازلية مرتفعة جداً تسمح بحماية الاشعة الضوئية من التداخل الخارجي.
4. المستقبل الضوئي: ومهمته تحويل الأشعة الضوئية المستقبلة إلى إشارة رقمية ذات معنى.
5. المصب: وهو المستقبل الإلكتروني الرقمي الذي يستفيد من المعلومة، في حالتنا هو مقياس معدل الخطأ.

تكون قيمة معدل الخطأ مساوية للصفر في الحالة المثالية كما هو موضح في الشكل (4-5).



الشكل(4-5). معدل الخطأ في الحالة المثالية

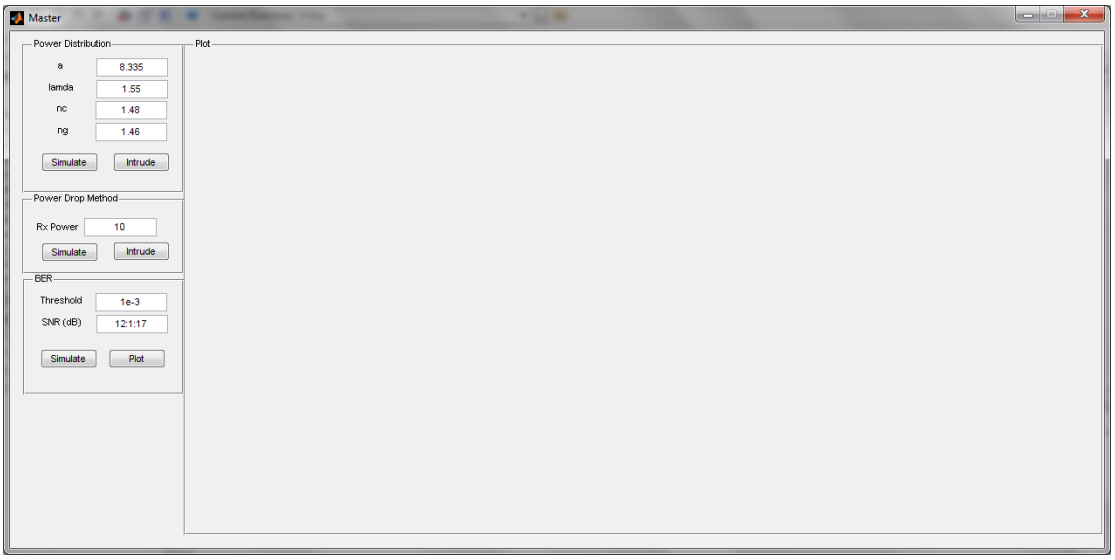
في حال وجود ضجيج ضمن قناة الاتصال ومن أجل نسبة إشارة للضجيج تساوي 5 dB فإن معدل الخطأ يساوي تقريباً 0.2 كما هو مبين بالشكل (4-6).



الشكل(4-6).المخطط الضندوقي لنظام الاتصالات في حالة وجود ضجيج

1.3. واجهة البرنامج:

بعد محاكاة كل جزء من مكونات النظام المقترح على حدا تم بناء واجهة للمستخدم بوساطة الماتلاب، يبين الشكل **(4-7)** واجهة المحاكاة للمشروع.



الشكل(4-7). واجهة المحاكاة

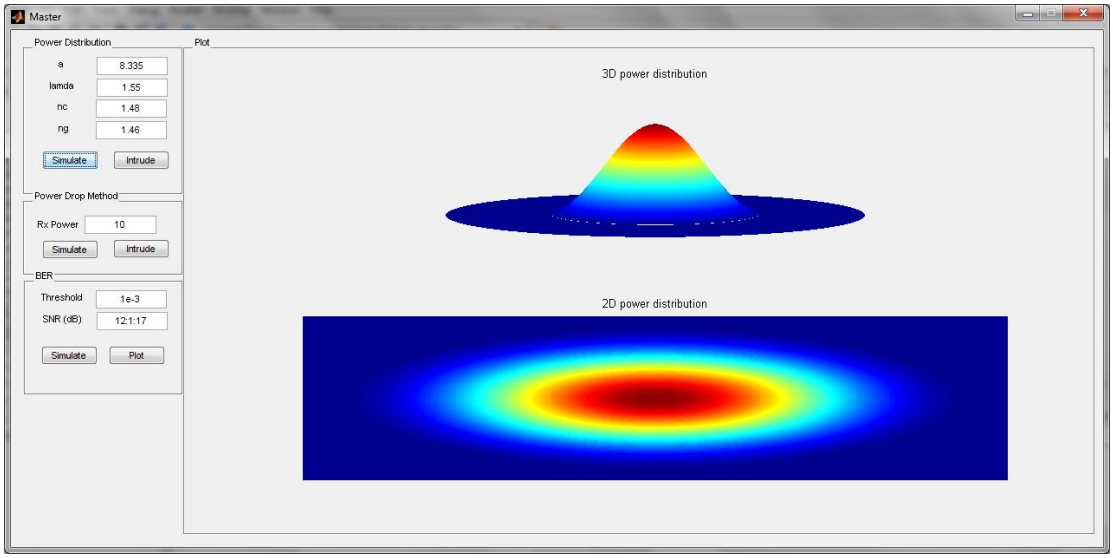
تتيح هذه الواجهة للمستخدم اختبار كل جزء من مكونات النظام على حدا، ويمكن استثمارها بالطريقة التالية:

- اختبار توزع الاستطاعة:

معاملات المحاكاة في هذه الحالة هي:

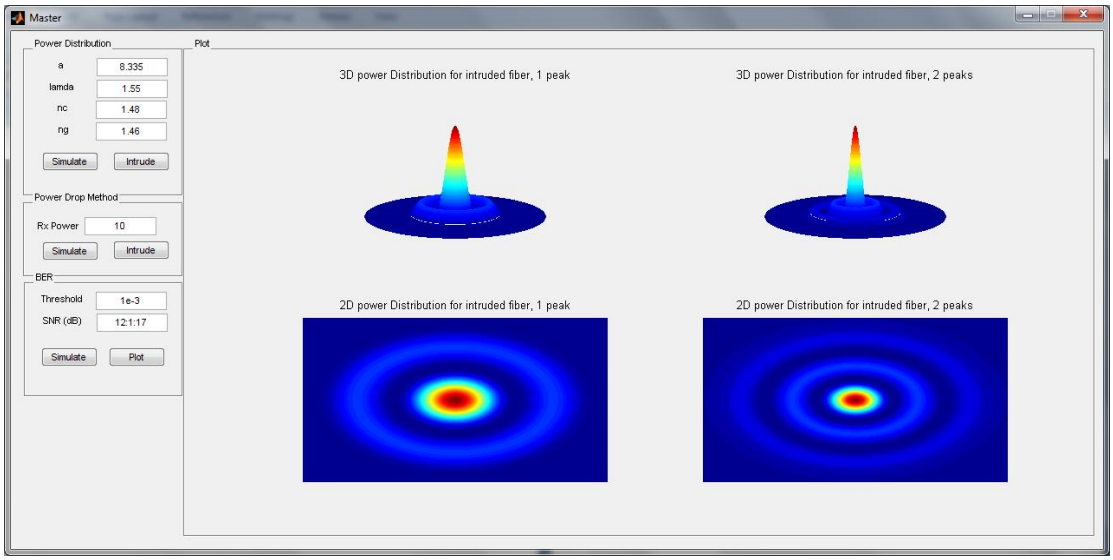
نصف قطر الليف (الرقم المدخل مقدرا بـ µm)، طول موجة الضوء المستخدم (الرقم المدخل مقدرا بـ µm)، قرينة انكسار اللب nc، قرينة انكسار الغلاف ng.

عند الضغط على الزر Simulate نحصل على النتيجة الموضحة بالشكل (4-8)



الشكل(4-8).التمثيل الفراغي لتوزع الاستطاعة

والتي توضح توزع الاستطاعة الضوئية في نهاية الليف، أما عند الضغط على الزر Intrude فالنتيجة التي نحصل عليها هي محاكاة لعملية اختراق الليف وهي موضحة في الشكل (4-9):

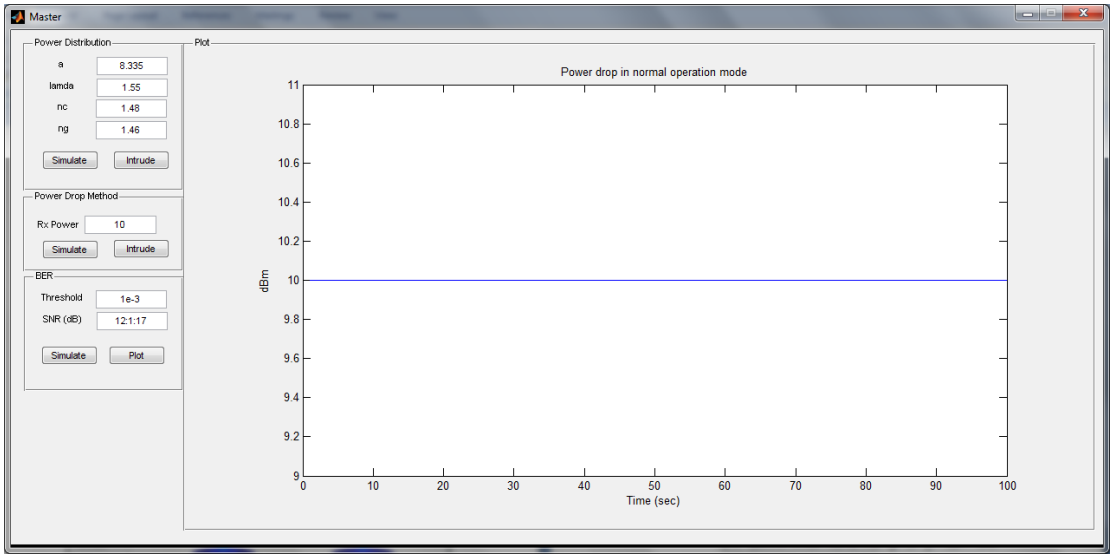


الشكل(4-9).توزع الاستطاعة في حال وجود اختراق

نلاحظ ظهور القمم المحلية العظمى لتوزع الاستطاعة والتي يعبر وجودها عن وجود اختراق في الليف.

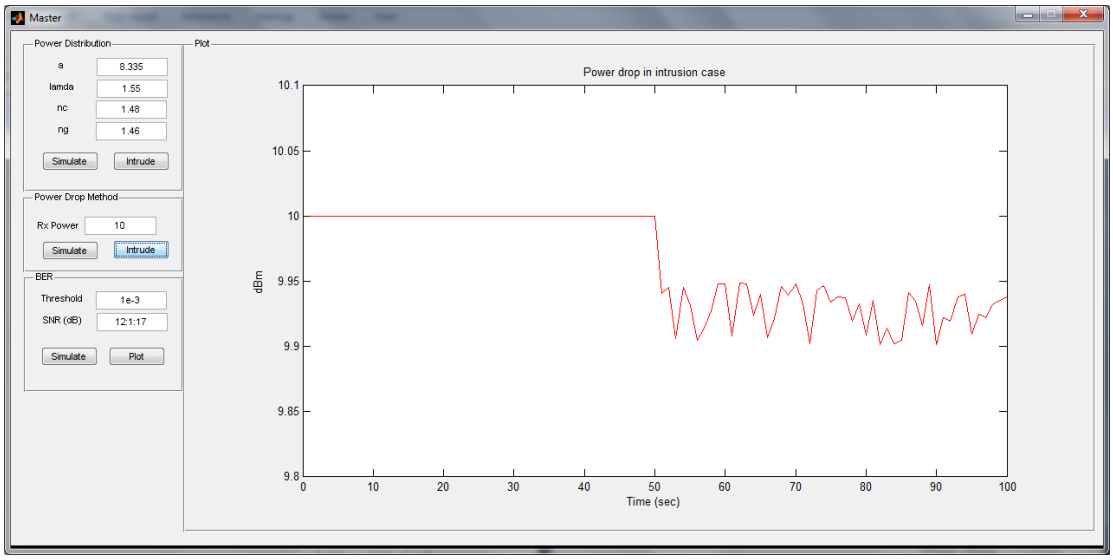
- اختبار انخفاض الاستطاعة:

يتم في هذه الحالة ادخال قيمة الاستطاعة المرسلة مقدرة بالـdBm، عند الضغط على الزر Simulate فإن النتيجة التي نحصل عليها هي محاكاة نمط العمل الطبيعي وفيه تكون قيمة الاستطاعة ثابتة وتساوي قيمة الاستطاعة المرسلة (نفترض في هذا الجزء ان الضياع في الليف الضوئي مساوياً للصفر) كما هو موضح بالشكل(4-10).



الشكل(4-10).الاستطاعة المستقبلة في حالة العمل الطبيعي

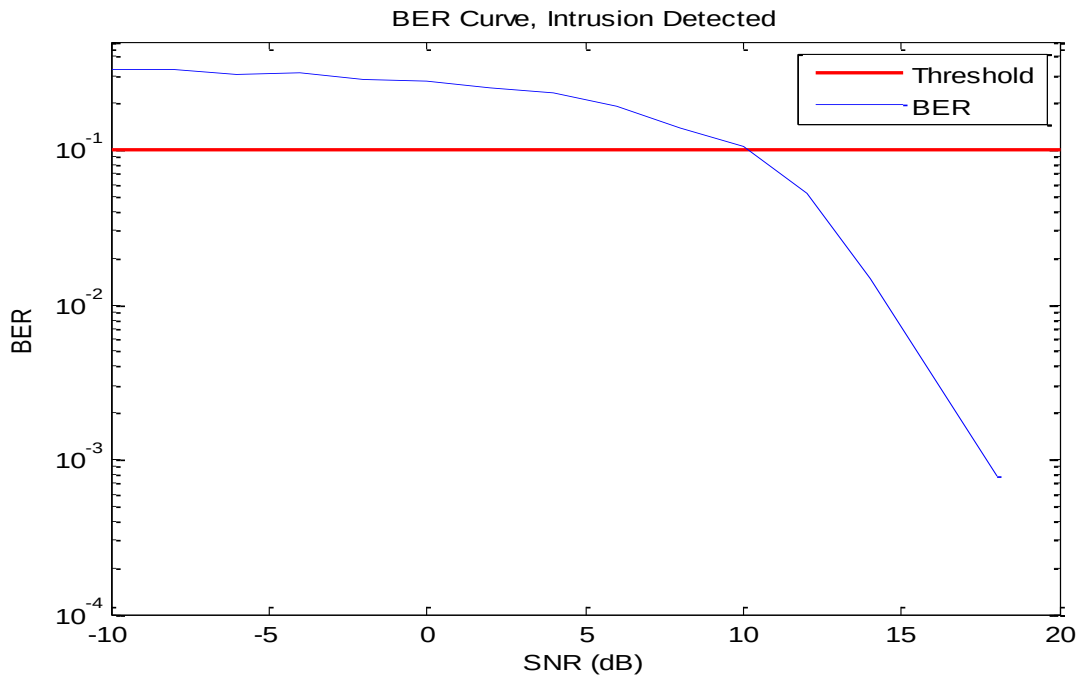
أما عند الضغط على الزر Intrude فتتم محاكاة قيمة الاستطاعة المستقبلة في حالة وجود اختراق ناتج عن ثني الليف، والنتيجة موضحة بالشكل(4-11):



الشكل(4-11).الاستطاعة المستقبلة في حال وجود اختراق

- اختبار معدل الخطأ:

لاختبار الجزء الخاص بمعدل الخطأ يجب ادخال قيمة العتبة التي يجب التنبيه عند تجاوز قيمة معدل الخطأ لها وكذلك يتم تحديد مجال تغير نسبة الاشارة للضجيج، النتيجة التي نحصل هي منحني أداء نظام الاتصالات الضوئي وهي مبينة في الشكل (4-12) من أجل قيم نسبة الاشارة للضجيج تنتمي للمجموعة:{10 ,9 , ... ,-8,-9,-10} يتم اولاً تنفيذ المحاكاة ثم رسم النتيجة



الشكل(4-12).منحي معدل الخطأ بدلالة نسبة الشارة للضجيج

- محاكاة جهاز OTDR:

عند صدور تنبيه عن وجود اختراق يتم استعمال جهاز OTDR لتحديد مكان الاختراق وهو ما تمت محاكاته في الجزء الأخير.

معاملات المحاكاة:

قرينة انكسار اللب nc.

معامل التخامد في الليف Betta مقدرا بـ dB/Km.

طول الليف Distance مقدرا بـ Km.

في حال وجود فقد في الاستطاعة P_{loss} ناجم عن الاختراق يتم إدخال هذه القيمة مقدرة بـ dB.

قيمة استطاعة الارسال P_t مقدرة بـ dB.

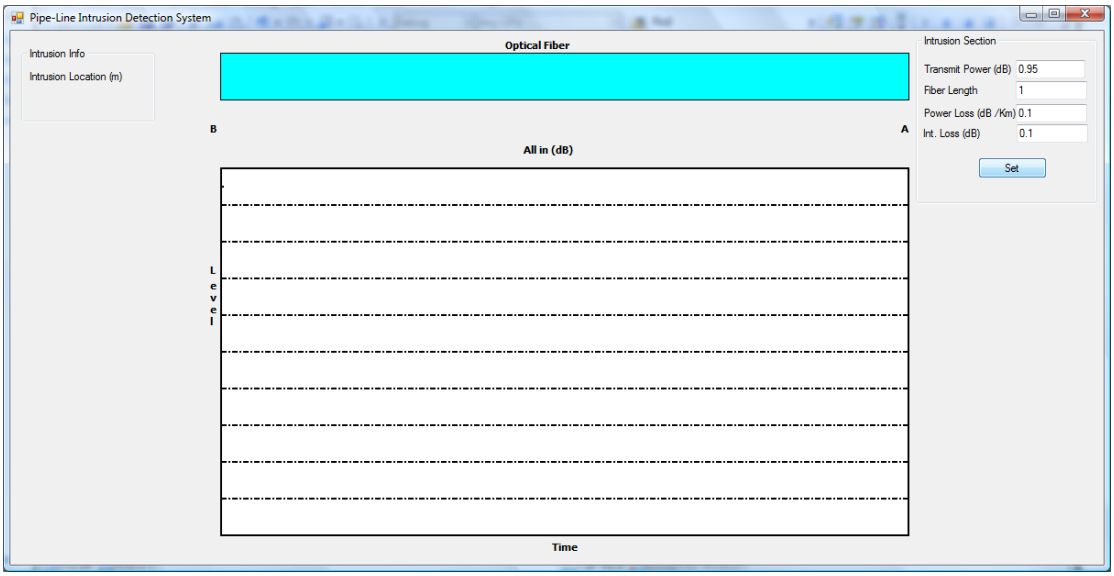
شرح المحاكاة:

يعمل جهاز OTDR بارسال نبضة ضوئية ثم قياس قيمة الاشارة المنعكسة، من تحديد الفاصل الزمني بين النبضة المرسلة والمستقبلة يتم قياس المسافة حيث ان الضوء ينتشر ضمن الليف بسرعة تساوي سرعة انتشار الضوء في الخلاء مقسومة على قرينة انكسار اللب، من قياس قيمة الاستطاعة يتم تحديد نوع الاختراق حيث ان الاستطاعة المستقبلة تحسب كما يلي:

$$P_r = P_t - 2*Distance*Betta - P_{loss}$$

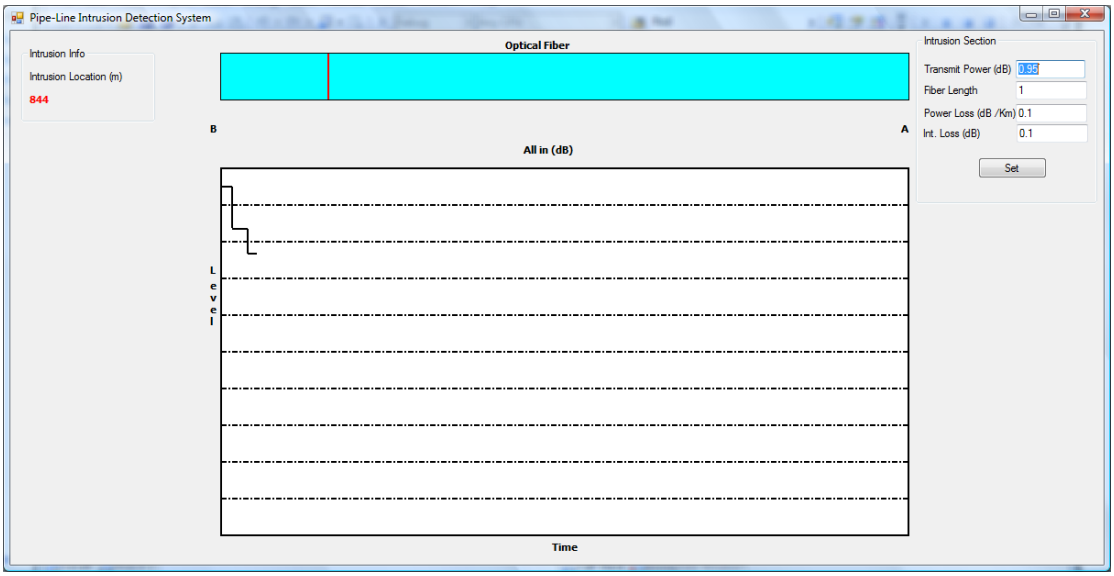
تم تنفيذ البرنامج باستخدام لغة البرمجة C# تتيح الواجهة للمستخدام اختبار نظام كشف الاختراق المصمم وتحديد موضع الاختراق.

يبين الشكل (4-13) واجهة البرنامج:



الشكل(4-13).واجهة البرنامج باستخدام لغة C#

يمكن تنفيذ الاختراق بكبستين متتاليتين على الليف، يظهر على الرسم تغيرات الاشارة المستقبلة كما يظهر مسافة الاختراق بواسطة جهاز OTDR كما هو موضح بالشكل التالي(4-14):



الشكل التالي(4-14).قياس المسافة باستخدام جهازOTDR

الخاتمة و الآفاق المستقبلية

عرضنا في هذه الأطروحة مشكلة الاختراقات التي تحدث ضمن شبكة الألياف الضوئية بأنواعها المختلفة. وعرضنا الحل لهذه المشكلة والمتلخص في بناء نظام إنذار يتضمن طرقاً ثلاثة في كشف وتحديد منطقة الاختراق، وقد قمنا بمحاكاة هذه الطرق الثلاث باستخدام الماتلاب ولغة البرمجة c# وحصلنا على نتائج جيدة في عمليات الكشف والتحديد لكل طريقة منها.

وفي تلخيص عملية بناء نظام الكشف فإن الطريقة الأولى تعتمد على قياس الاستطاعة الضوئية المستقبلة في نهاية الليف ومن ثم مقارنتها بقيمة الاستطاعة المرسلة، وعند تغير مستوى الاستطاعة المستقبلة فإننا نستنتج وجود مشكلة اختراق في منطقة ما نستطيع تحديدها تبعاً لعلاقة الاستطاعة المستقبلة التي شرحناها سابقاً.

أما الطريقة الثانية فتعتمد على قياس التوزع المكاني للاستطاعة الضوئية المستقبلة التي كما تحدثنا فإنها تتبع توزع بيسل في الحالة الطبيعية ، ولكن في حال حدوث اختراق فإن هذا التوزع يتغير ونحصل على أشكال أخرى من توزع بيسل التي ندعوها بالتوزيع المعدل. وفي هذه الطريقة لانستطيع قياس أو تحديد المنطقة التي تمت فيها عملية الاختراق.

وفي الطريقة الثالثة نعتمد على قياس الخطأ BER المرتكب بين المرسل والمستقبل. حيث أنه يمكن تحديد منحني الخطأ للنظام الموجود بدايةً عند تركيبه بقياسه تجريبياً، ومن ثم تحديد عتبة الضجيج التي نقول عندها أنه تم اختراق الشبكة.

وكخطوات مستقبلية في هذا الموضوع يمكننا أن نقوم بـ :

- تنفيذ النظام المقترح واختباره عملياً.
- تطبيق طرق حماية على المستوى المنطقي للنظام (كخوارزميات التشفير)، والتي تفيد في زيادة مناعة نظام الاتصالات ضد عمليات الاختراق. حينئذ وعند حصول الاختراق الفيزيائي فإن المعلومات المرسلة تكون محمية ومشفرة ولن تستطيع الجهة المخترقة الحصول على هذه المعلومات ببساطة.

الملحق 1 : برنامج المحاكاة

```
function varargout = Master(varargin)
% MASTER M-file for Master.fig
%      MASTER, by itself, creates a new MASTER or
raises the existing
%      singleton*.
%
%      H = MASTER returns the handle to a new MASTER
or the handle to
%      the existing singleton*.
%
%
MASTER('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in MASTER.M with the
given input arguments.
%
%      MASTER('Property','Value',...) creates a new
MASTER or raises the
%      existing singleton*.  Starting from the left,
property value pairs are
%      applied to the GUI before Master_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value
makes property application
%      stop.  All inputs are passed to
Master_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.
Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Master

% Last Modified by GUIDE v2.5 19-Mar-2011 18:12:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton,
...
```

```
                   'gui_OpeningFcn',
@Master_OpeningFcn, ...
                   'gui_OutputFcn',
@Master_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Master is made visible.
function Master_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
% varargin   command line arguments to Master (see
VARARGIN)

% Choose default command line output for Master
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Master wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the
command line.
function varargout = Master_OutputFcn(hObject,
eventdata, handles)
```

```
% varargout  cell array for returning output args
(see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit5 as text
%        str2double(get(hObject,'String')) returns
contents of edit5 as a double


% --- Executes during object creation, after setting
all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit6 as text
%        str2double(get(hObject,'String')) returns
contents of edit6 as a double


% --- Executes during object creation, after setting
all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global Threshold SNR Ber IntrusionDetected
warning off
%softtdm;
Threshold = str2double(get(handles.edit5,'String'));
SNR =
evalin('base',strcat('[',get(handles.edit6,'String'),
'];'));
Ber = zeros(1,length(SNR));
```

```
IntrusionDetected = 0;
for i = 1:length(SNR),
    snr_dB = SNR(i);
    evalin('base',strcat('snr_dB =
',num2str(snr_dB),';'));

evalin('base',strcat('sim(',''''Simple_Noise''',');'))
;
    in = evalin('base','in;');
    out = evalin('base','out;');
    in = reshape(in,1,20*193);
    out = reshape(out,1,20*193);
    [temp,Ber(i)] = biterr(in,out);
    if Ber(i) >= Threshold
        IntrusionDetected = 1;
    end
end
msgbox('Simulation Finished !!');



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit1 as text
%        str2double(get(hObject,'String')) returns
contents of edit1 as a double


% --- Executes during object creation, after setting
all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit2 as text
%        str2double(get(hObject,'String')) returns
contents of edit2 as a double


% --- Executes during object creation, after setting
all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
edit3 as text
%        str2double(get(hObject,'String')) returns
contents of edit3 as a double


% --- Executes during object creation, after setting
all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit4 as text
%        str2double(get(hObject,'String')) returns
contents of edit4 as a double


% --- Executes during object creation, after setting
all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
a=str2double(get(handles.edit1,'String'));
lambda=str2double(get(handles.edit2,'String'));
nc=str2double(get(handles.edit3,'String'));
ng=str2double(get(handles.edit4,'String'));
v=(2*pi*a/lambda)*sqrt(nc^2-ng^2);
umax=v;umin=0;wmax=v;wmin=0;
u=[0:0.1:v];
Su=size(u);
for m=1:Su(2);
    J(m)=u(m)*besselj(1,u(m))/besselj(0,u(m));
end
for m=1:Su(2);
    w(m)=sqrt(v^2-u(m)*u(m));
    K(m)=w(m)*besselk(1,w(m))/besselk(0,w(m));
end
 %plot(u,J);hold on;plot(w,K);
u01=2.1845;neff01=sqrt(nc^2-(u01/(2*pi*a/lambda))^2);
u02=4.9966;neff02=sqrt(nc^2-(u02/(2*pi*a/lambda))^2);
u03=7.7642;neff03=sqrt(nc^2-(u03/(2*pi*a/lambda))^2);
w01=sqrt(v^2-u01^2);
w02=sqrt(v^2-u02^2);
w03=sqrt(v^2-u03^2);
rc=[0:0.1:a];
Src=size(rc);
for m=1:Src(2);
    psic01(m)=besselj(0,u01*rc(m)/a)/besselj(0,u01);
end
```

```
rg=[a:0.1:2*a];
Srg=size(rg);
for m=1:Srg(2);
    psig01(m)=besselk(0,w01*rg(m)/a)/besselk(0,w01);
end
psic01t=psic01';
psig01t=psig01';
psic01t = psic01t(end:-1:1);
psig01t=psig01t(end:-1:1);
     PSIC01=repmat(psic01t,1,500);
     PSIG01=repmat(psig01t,1,500);
[Xc,Yc,Zc]=polar3d(PSIC01.^2/max(max(PSIC01))^2,0,2*p
i,0,a,2,'off');
[Xg,Yg,Zg]=polar3d(PSIG01.^2/max(max(PSIC01))^2,0,2*p
i,a,2*a,2,'off');

subplot(2,1,1,'Parent',handles.uipanel2);
surf(Xc,Yc,Zc); hold on;
surf(Xg,Yg,Zg);axis([-20 20 -20 20]);shading interp;
axis off;title('3D power distribution');
subplot(2,1,2,'Parent',handles.uipanel2);
pcolor(Xc,Yc,Zc);hold on;pcolor(Xg,Yg,Zg);axis([-10
10 -10 10]);shading interp;axis off;title('2D power
distribution');
%colorbar

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
a=str2double(get(handles.edit1,'String'));
lambda=str2double(get(handles.edit2,'String'));
nc=str2double(get(handles.edit3,'String'));
ng=str2double(get(handles.edit4,'String'));
v=(2*pi*a/lambda)*sqrt(nc^2-ng^2);
umax=v;umin=0;wmax=v;wmin=0;
u=[0:0.1:v];
Su=size(u);
for m=1:Su(2);
    J(m)=u(m)*besselj(1,u(m))/besselj(0,u(m));
end
for m=1:Su(2);
    w(m)=sqrt(v^2-u(m)*u(m));
```

```
    K(m)=w(m)*besselk(1,w(m))/besselk(0,w(m));
end
 %plot(u,J);hold on;plot(w,K);
u01=2.1845;neff01=sqrt(nc^2-(u01/(2*pi*a/lambda))^2);
u02=4.9966;neff02=sqrt(nc^2-(u02/(2*pi*a/lambda))^2);
u03=7.7642;neff03=sqrt(nc^2-(u03/(2*pi*a/lambda))^2);
w01=sqrt(v^2-u01^2);
w02=sqrt(v^2-u02^2);
w03=sqrt(v^2-u03^2);
rc=[0:0.1:a];
Src=size(rc);
for m=1:Src(2);
    psic02(m)=besselj(0,u02*rc(m)/a)/besselj(0,u02);
end
rg=[a:0.1:2*a];
Srg=size(rg);
for m=1:Srg(2);
    psig02(m)=besselk(0,w02*rg(m)/a)/besselk(0,w02);
end
psic02t=psic02';
psig02t=psig02';
psic02t=psic02t(end:-1:1);
psig02t=psig02t(end:-1:1);
     PSIC02=repmat(psic02t,1,500);
     PSIG02=repmat(psig02t,1,500);
[Xc,Yc,Zc]=polar3d(PSIC02.^2/min(min(PSIC02))^2,0,2*p
i,0,a,2,'off');
[Xg,Yg,Zg]=polar3d(PSIG02.^2/min(min(PSIC02))^2,0,2*p
i,a,2*a,2,'off');

subplot(2,2,1,'Parent',handles.uipanel2);
surf(Xc,Yc,Zc); hold on;surf(Xg,Yg,Zg);axis([-20 20 -
20 20]);shading interp;axis off;title('3D power
Distribution for intruded fiber, 1 peak');
subplot(2,2,3,'Parent',handles.uipanel2);
pcolor(Xc,Yc,Zc);hold on;pcolor(Xg,Yg,Zg);axis([-10
10 -10 10]);shading interp;axis off;title('2D power
Distribution for intruded fiber, 1 peak');


v=(2*pi*a/lambda)*sqrt(nc^2-ng^2);
umax=v;umin=0;wmax=v;wmin=0;
u=[0:0.1:v];
Su=size(u);
for m=1:Su(2);
    J(m)=u(m)*besselj(1,u(m))/besselj(0,u(m));
end
```

```
for m=1:Su(2);
    w(m)=sqrt(v^2-u(m)*u(m));
    K(m)=w(m)*besselk(1,w(m))/besselk(0,w(m));
end
 %plot(u,J);hold on;plot(w,K);
u01=2.1845;neff01=sqrt(nc^2-(u01/(2*pi*a/lambda))^2);
u02=4.9966;neff02=sqrt(nc^2-(u02/(2*pi*a/lambda))^2);
u03=7.7642;neff03=sqrt(nc^2-(u03/(2*pi*a/lambda))^2);
w01=sqrt(v^2-u01^2);
w02=sqrt(v^2-u02^2);
w03=sqrt(v^2-u03^2);
rc=[0:0.1:a];
Src=size(rc);
for m=1:Src(2);
    psic03(m)=besselj(0,u03*rc(m)/a)/besselj(0,u03);
end
rg=[a:0.1:2*a];
Srg=size(rg);
for m=1:Srg(2);
    psig03(m)=besselk(0,w03*rg(m)/a)/besselk(0,w03);
end
psic03t=psic03';
psig03t=psig03';
psic03t = psic03t(end:-1:1);
psig03t=psig03t(end:-1:1);
     PSIC03=repmat(psic03t,1,500);
     PSIG03=repmat(psig03t,1,500);
[Xc,Yc,Zc]=polar3d(PSIC03.^2/max(max(PSIC03))^2,0,2*p
i,0,a,2,'off');
[Xg,Yg,Zg]=polar3d(PSIG03.^2/max(max(PSIC03))^2,0,2*p
i,a,2*a,2,'off');

subplot(2,2,2,'Parent',handles.uipanel2);
surf(Xc,Yc,Zc); hold on;surf(Xg,Yg,Zg);axis([-20 20 -
20 20]);shading interp; axis off;title('3D power
Distribution for intruded fiber, 2 peaks');
subplot(2,2,4,'Parent',handles.uipanel2);
pcolor(Xc,Yc,Zc);hold on;pcolor(Xg,Yg,Zg);axis([-10
10 -10 10]);shading interp;axis off;title('2D power
Distribution for intruded fiber, 2 peaks');



% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton5 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global Threshold SNR Ber IntrusionDetected
figure
semilogy(SNR,Threshold*ones(1,length(SNR)),'-
r','LineWidth',2);
hold on
semilogy(SNR,Ber,'LineWidth',1);
ylim([1e-4 0.5]);
legend('Threshold','BER');
tit = 'BER Curve';
if IntrusionDetected == 1
    tit = strcat(tit,', Intrusion Detected');
else
    tit = strcat(tit,', No Intrusion Detected');
end
title(strcat(tit)),xlabel('SNR (dB)'),ylabel('BER');



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit9 as text
%        str2double(get(hObject,'String')) returns
contents of edit9 as a double


% --- Executes during object creation, after setting
all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
Tx = str2double(get(handles.edit9,'String'));
figure, plot(Tx*ones(1,100)),ylabel('Power
(dB)'),title('No Intrusion Deected');

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
Tx = str2double(get(handles.edit9,'String'));
y = (rand(1,50) - 0.5)/10 + (Tx - 0.2)*ones(1,50);
figure,plot([Tx*ones(1,50) y]),ylabel('Power
(dB)'),ylim([Tx-1 Tx+1]),title('Intrusion Detected');



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit10 as text
%        str2double(get(hObject,'String')) returns
contents of edit10 as a double
```

```
% --- Executes during object creation, after setting
all properties.
function edit10_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit11 as text
%        str2double(get(hObject,'String')) returns
contents of edit11 as a double


% --- Executes during object creation, after setting
all properties.
function edit11_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit12 as text
%        str2double(get(hObject,'String')) returns
contents of edit12 as a double


% --- Executes during object creation, after setting
all properties.
function edit12_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit13 as text
%        str2double(get(hObject,'String')) returns
contents of edit13 as a double


% --- Executes during object creation, after setting
all properties.
function edit13_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

n = str2double(get(handles.edit10,'String'));
Betta = str2double(get(handles.edit11,'String'));
IntrosionDistance =
str2double(get(handles.edit12,'String'));
IntrosionLoss =
str2double(get(handles.edit13,'String'));
Pt = str2double(get(handles.edit15,'String'));

Bitrate = 1;
```

```
c = 3*10^8;
Speed = c/n;              % speed of light

PulseWidth = 1/(Bitrate * 10^6);
Ts = PulseWidth/100;
Pi = Pt - IntrosionDistance*Betta - IntrosionLoss;
Pr = Pi - IntrosionDistance*Betta;
Tao  = 2*IntrosionDistance*1000/Speed;
N  = ceil(Tao/Ts);

Tx = 10^(Pt/10)*ones(1,10); % Transmited Bit
Rx = 10^(Pr/10)*ones(1,10); % Recieved Bit

N1 = 2*N;
Tx = [zeros(1,N/2) Tx zeros(1,N1 - length(Tx))];
Rx = [zeros(1,3*N/2) Rx zeros(1,N - length(Rx))];
t = 0:Ts:(N1-1 + N/2)*Ts;
figure;
plot(t,Tx);
hold on
plot(t,Rx,'r');
legend('Transmitted Puls','Detected Pulse');
xlabel('Time (sec)');
ylabel('dBm');
title(['intrusion detected at D = '
num2str(IntrosionDistance) ' Km']);

function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit15 as text
%        str2double(get(hObject,'String')) returns
contents of edit15 as a double


% --- Executes during object creation, after setting
all properties.
function edit15_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit15 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit16 as text
%        str2double(get(hObject,'String')) returns
contents of edit16 as a double


% --- Executes during object creation, after setting
all properties.
function edit16_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
power = str2double(get(handles.edit16,'String'));
subplot(1,1,1,'Parent',handles.uipanel2);
plot(power*ones(1,100)),ylabel('dBm'),xlabel('Time
(sec)'),title('Power drop in normal operation mode');

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
power = str2double(get(handles.edit16,'String'));
subplot(1,1,1,'Parent',handles.uipanel2);

plot([power*ones(1,50) ((power -
0.1)*ones(1,50)+0.05*rand(1,50))],'r');
ylim([power-0.2 power+0.1]);
ylabel('dBm'),xlabel('Time (sec)'),title('Power drop
in intrusion case');
```

OTDR Prog:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Media;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        #region Properties
        // Axes
        public int top = 150;
        public int left = 200;
        public int width = 750;
        public int height = 400;
        public string Wav =
"C:\\WINDOWS\\Media\\Windows Error.wav";
        // Pipe Line
        public int pipe_top = 25;
        public int pipe_left = 200;
        public int pipe_width = 750;
        public int pipe_height = 50;
        public bool Clicked = false;
        public Point pnt_Clicked = new
Point();
        public double FiberLength = 1;// Km
        public double TransmitPower = 0.95;//m
        public double PowerLoss = 0.1;// dB/Km
        public double IntLoss = 0.1; // dB
        // Data Acquisition
        public double[] Data;
        public int Data_idx = 0;
```

```
        public double axes_y = 0;
        public double axes_y_d = 0;

        public Graphics gs;

        public bool IntrusionDetected = false;

        public double Threshold = 0;
        public bool Flash = false;

        public SoundPlayer pl;

        public ToolTip ThresholdTip;
        #endregion

        public Form1()
        {
            InitializeComponent();
            Data = new double [width];
            Data = new double[width];
            timer1.Interval = 1000;
            timer1.Start();
            Threshold = 0.5;
            axes_y = 1;
            axes_y_d = 0;
            pl = new SoundPlayer(Wav);
        }

        private void Form1_Load(object sender,
EventArgs e)
        {
            this.Focus();

            ThresholdTip = new ToolTip();
            ThresholdTip.AutoPopDelay = 5000;
            ThresholdTip.InitialDelay = 1000;
            ThresholdTip.ReshowDelay = 1000;
        }
```

```
        private void Form1_Paint(object
sender, PaintEventArgs e)
        {
            gs =
Graphics.FromHwnd(this.Handle);
            Pen myPen = new Pen(Color.Black,
2);
            gs.FillRectangle(Brushes.White,
left, top, width, height);
            gs.DrawRectangle(myPen, left, top,
width, height);
            myPen.DashStyle =
System.Drawing.Drawing2D.DashStyle.DashDot;
            int y = 0;
            int x = left + width;
            for (int i = 0; i < 10; i++)
            {
                y = top + i * 40;
                gs.DrawLine(myPen, left, y, x,
y);
            }
            //myPen = new Pen(Color.Red, 2);
            //gs.DrawLine(myPen, left, top +
(int)((axes_y - Threshold ) * height / (axes_y
- axes_y_d)), x, top + (int)((axes_y -
Threshold) * height / (axes_y - axes_y_d)));
            Font font = new Font("Tahoma",10);
            //gs.DrawString("Intrusion Level",
font, Brushes.Red, x / 2, top + (int)((axes_y
- Threshold) * height / (axes_y - axes_y_d)));
            //gs.DrawString("Intrusion Area",
font, Brushes.Red, x / 2, top + (int)((axes_y
- Threshold) * height / (axes_y - axes_y_d)) +
50);
            //gs.DrawString("Green Area",
font, Brushes.Green, x / 2, top +
(int)((axes_y - Threshold) * height / (axes_y
- axes_y_d)) - 50);
            myPen = new Pen(Color.Black, 2);
```

```
            //gs.DrawImage(Mic, new
Rectangle(pipe_left, pipe_top + pipe_height,
10, 20));
            //gs.DrawImage(Mic, new
Rectangle(pipe_left + pipe_width - 10,
pipe_top + pipe_height, 10, 20));

            gs.DrawRectangle(myPen, pipe_left,
pipe_top, pipe_width , pipe_height);
            gs.FillRectangle(Brushes.Aqua,
pipe_left, pipe_top, pipe_width, pipe_height);
            // Draw Intrusion Point
            myPen = new Pen(Color.Red, 2);
            if (Clicked)
            {
                gs.DrawLine(myPen,
pnt_Clicked.X, pipe_top, pnt_Clicked.X,
pipe_height + pipe_top);
            }
            // Draw Acquisition Data
            if (Data_idx > 0)
            {
                Point[] temp = new
Point[Data_idx];
                for (int i = 0; i < Data_idx;
i++)
                {
                    temp[i].Y = (int)((axes_y
- Data[i]) * height / (axes_y - axes_y_d));
                    temp[i].Y += top;
                    temp[i].X = i+left;
                }
                if (Data_idx > 2)
                {
                    myPen = new
Pen(Color.Black, 2);
                    gs.DrawLines(myPen, temp);
                    myPen = new
Pen(Color.Blue, 2);
                }
            }
```

```
        }

        private void timer1_Tick(object
sender, EventArgs e)
        {
            Random r = new Random();
            lock (Data)
            {
                if (Data_idx == width)
                {
                    Data = new double[width];
                    Data_idx = 0;
                }


                if (Clicked)
                {
                    double dist = FiberLength
- (pnt_Clicked.X - pipe_left) * FiberLength /
pipe_width;

                    Data[Data_idx] =
TransmitPower - IntLoss - dist * PowerLoss;

                    //// Calculate Distance
                    IntrusionDetected = false;
                    if (Data[Data_idx] <=
Threshold)
                    {
                        IntrusionDetected =
true;
                    }
                }
                else
                {
                    Data[Data_idx] =
TransmitPower;
                }


                Data_idx++;
```

```
            }
            Form1_Paint(null, null);
        }

        private void button_Noise_Click(object
sender, EventArgs e)
        {
            Data = new double[width];
            Data_idx = 0;
            FiberLength =
(double)Double.Parse(textBox_Distance.Text);
            TransmitPower =
(double)Double.Parse(textBox_Level.Text);
            PowerLoss =
(double)Double.Parse(textBox_PwrLss.Text);
            IntLoss =
(double)Double.Parse(textBox_IntLss.Text);
            Form1_Paint(null, null);
        }

        private void Form1_MouseMove(object
sender, MouseEventArgs e)
        {

            if ((e.X > pipe_left) && (e.X <
(pipe_width + pipe_left)) && (e.Y > pipe_top)
&& (e.Y < (pipe_height + pipe_top)))
            {


                //
            }

            int threshold_display =
(int)((axes_y - Threshold) * height / (axes_y
- axes_y_d)) + top;
            if (((threshold_display - 5) <
e.Y) && (e.Y < (threshold_display + 5)))
            {
                ThresholdTip.SetToolTip(this,
"Threshold");
```

```
            }
            else
            {
                ThresholdTip.RemoveAll();
                ThresholdTip.Hide(this);
            }
        }

        private void
Form1_MouseDoubleClick(object sender,
MouseEventArgs e)
        {
            double dist = -1;
            if ((e.X > pipe_left) && (e.X <
(pipe_width + pipe_left)) && (e.Y > pipe_top)
&& (e.Y < (pipe_height + pipe_top)))
            {

                Clicked = true;
                dist = FiberLength * 1000 -
(e.X - pipe_left) * FiberLength * 1000 /
pipe_width;
                pnt_Clicked = e.Location;
                Graphics gs =
Graphics.FromHwnd(this.Handle);
                Pen myPen = new Pen(Color.Red,
2);

//gs.FillRectangle(Brushes.Aqua, pipe_left,
pipe_top, pipe_width, pipe_height);
                gs.DrawLine(myPen,
pnt_Clicked.X, pipe_top, pnt_Clicked.X,
pipe_height + pipe_top);
            }
            if (dist != -1)
                label_Mouse.Text =
dist.ToString("F00");
            else
                label_Mouse.Text = "";
        }
```

```
        private void Form1_MouseClick(object
sender, MouseEventArgs e)
        {
            if (e.Button ==
MouseButtons.Right)
            {
                Clicked = false;
                Form1_Paint(null, null);
                label_Mouse.Text = "";
            }

        }

        private void comboBox1_KeyPress(object
sender, KeyPressEventArgs e)
        {
            e.Handled = true;
        }

        private void button2_Click(object
sender, EventArgs e)
        {
        }


    }
}
```

مقدمة:

تطورت في الآونة الأخيرة التقنيات المستخدمة في أنظمة الاتصالات ومعالجة الاشارة الرقمية بشكل كبير ولمجاراة هذه التقنيات تطورت بالمقابل أدوات التنفيذ من معالجات إشارة رقمية *DSP* والـ *FDGA* وغيرها، وبالتالي أصبح زمن التصميم والتنفيذ كبيرا كما أن الكلفة أصبحت مرتفعة جدا (سعر المتحكم الصغري بحدود 150 ليرة سورية في حين أن سعر معالج الاشارة الرقمية بحدود 15000 ليرة سورية)، هذا التطور تطلب استخدام تقنيات حاسوبية جديدة لم تكن مستخدمة من قبل تساعد المهندس في عملية التصميم والاختبار قبل الانتقال إلى مرحلة التنفيذ العملي، هذه التقنيات هي النمذجة والمحاكاة وهي ما سندرسه في هذا الفصل.

النمذجة: هي عملية اقتراح شكل الموديل الحاكم لعمل الجملة، وهذا الموديل يمكن اشتقاقه من:

- المعادلات التفاضلية التي يمكن اشتقاقها فيزيائياً بحسب نوع الجملة.
- المعادلات التفاضلية الجزئية التي تكون صحيحة حول نقطة عمل الجملة، عادة ما تكون الجملة خاضعة لمعادلات تفاضلية غير خطية أو لايمكن اشتقاقها بسهولة، بالتالي نلجأ لعملية التقدير.
- الخبرة الناجمة عن مراقبة عمل الجملة.

قد يكون هذا الموديل محدد *Deterministic* أو احصائي *Stochastic*.

ينتج عن مرحلة النمذجة موديلات من الأشكال: $\frac{k}{1+TP}$ وهو نموذج استجابة النظام من الدرجة الأولى، وقد يكون من درجات أعلى (حسب تعقيد النظام).

يتم بعد إيجاد النموذج إجراء عملية مطابقة له، وتهدف هذه العلمية إلى حساب قيم متوسطات الموديل بحيث ينطبق خرج الموديل مع الخرج الحقيقي.

توجد عدة طرق لاستخراج النموذج من أشهرها طريقة التربيعات الصغرى *Minimum Least Square*، والتي تعتبر الاساس لأغلب طرق النمذجة.

أشهر المحاكيات:

PSpice: وهو برنامج محاكاة للدارات الكهربائية والالكترونية، يتيح لنا إجراء محاكاة لبعض الظواهر التي لا نستطيع في الواقع تحديدها كالأثر الحراري على العناصر.

Pretuse: وهو برنامج يقوم بعمل محاكاة لعمل المتحكم الصغري *Microcontroller*، وذلك بعد تحميل ملف البرنامج عليه.

Microwave Office: وهو برنامج يقوم بعمل محاكاة لانتشار الامواج الكهرطيسية وكذلك للدارات المكروية.

الماتلاب *MATLAB*: وهو عبارة عن برنامج ضخم يسمح لنا بإجراء محاكاة لمختلف أنواع النظم وفي مجالات اختصاصية كثيرة، وتتم المحاكاة إما عن طريق البرمجة (كتابة كود) او باستخدام *Simulink* وفيه يتم بناء النظام بشكل صناديق متصلة مع بعضها.

المحاكاة:

نقدم فيما يلي لأحد أهم الأسس الرياضية لعملية المحاكاة والتي تعتمد بشكل مباشر على طريقة مونت كارلو.تعتمد طرق مونت كارلو في المحاكاة والتقدير على ما يسمى بلعبة الحظ وتقسم إلى:

محاكاة مونت كارلو Monte Carlo Simulation: وهي المحاكاة التي يتم فيها تقدير احد معاملات النظام (مثل معدل الخطأ بالبت) باستخدام مقدرات مونت كارلو Monte Carlo Estimator.

مقدرات مونت كارلو: وهي طريقة في إيجاد تقدير لقيمة متحول ما وذلك إما بتنفيذ عمليات إحصائية او تجارب عشوائية.

التردد النسبي Relative Frequency:

تعتمد مُقدرات مونت كارلو على التردد النسبي في تقدير احتمال حدث ما، بالعودة إلى نظرية الاحتمالات فإن التجربة العشوائية هي التجربة التي يكون الخرج فيها لا يمكن توقعه ولكن يمكن تحديد احتمال الحصول على قيمة ما في خرج التجربة.

نقصد بحدث مجموعة جزئية من قيم الخرج الممكنة لتجربة عشوائية ما، فمثلاً يهمنا في أنظمة الاتصالات احتمال حدوث خطأ لذلك فالحدث هنا هو حدوث خطأ في الاستقبال اي إرسال البت 1 واستقبال البت 0 وبالعكس.

بعد هذه المقدمة الموجزة عن الاحتمالات والأحداث نقدم لمقدرات مونت كارلو والتي تعتمد على تكرار تجربة عشوائية ما عدد من المرات N، بالتعريف فإن تقدير احتمال وقوع الحدث A هو النسبة N_A/N حيث N_A عدد مرات وقوع الحدث A عند تكرار التجربة العشوائية، النسبة السابقة هي تقدير ونعلم ان احتمال وقوع هذا الحدث هو:

$$\Pr(A) = \lim_{N\to\infty}\left(\frac{N_A}{N}\right)$$

لذلك قلنا تقدير للاحتمال، ويكون خطأ التقدير أصغر كلما كان عدد مرات تكرار التجربة أكبر، لذلك إذا أردنا تقدير احتمال الخطأ بالبت على سبيل المثال يجب علينا إرسال (محاكاة) عدد كبير من البتات وكلما كان هذا العدد كبيراً كلما كان خطأ التقدير أقل أي تقترب قيمة المقدر من القيمة الحقيقية لهذا المعامل (معدل الخطأ بالبت). نرمز لمقدرات مونت كارلو بالرمز $\widehat{\Pr}(A)$، وتجدر الملاحظة هنا أنه ومن أجل قيمة محددة لعدد التجارب فإن N_A هو متحول عشوائي وبالتالي يكون $\widehat{\Pr}(A)$ متحول عشوائي أيضاً، تحدد الخواص الإحصائية لهذا المتحول العشوائي دقة عملية التقدير وبالتالي تعبر عن جودة المحاكاة.

المقدرات غير المزاحة والمتماسكة:

لتكتسب مقدرات مونت كارلو أهمية يجب تحقيق العديد من الخواص منها:

غير مزاح Unbiased: أي انه إذا كان $\hat{A}$ تقدير للمعامل (أو الحدث) A نريد ان يكون $E\{\hat{A}\} = A$

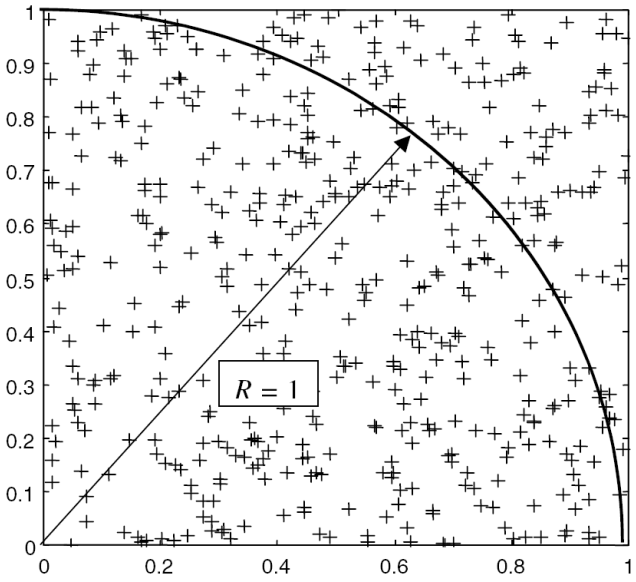
أي انه بالمتوسط فإن القيمة الحقيقية للمعامل يتم الحصول عليها، كما أنه وعند تكرار التجارب عدد كبير من المرات نريد أن تكون القيم المقدرة ذات تشتت صغير، فإذا تحقق هذين الشرطين عندئذ تعطي عملية التقدير عدة قيم قريبة من القيمة الحقيقة وتوزعها الهندسي حول القيمة الحقيقة يكون ذو مساحة صغيرة (تصغر هذه المسافة بزيادة عدد التجارب)، أي ان: $\lim_{N\to\infty} {\sigma_{\hat{A}}}^2 = 0$.

نعرف في حالة المقدرات غير المزاحة والمتماسكة خطأ التقدير بالعلاقة: $e = \hat{A} - A$ ويكون الخطأ في هذه الحالة متحول عشوائي بمتوسط معدوم وبتشتت يقترب من الصفر، لكن وفي أغلب الحالات يكون هذا الاقتراب بطيئا أي اننا بحاجة لعدد كبير من التجارب.

نوضح عمل مقدرات مونت كارلو بالمثال التالي والذي نريد فيه إيجاد قيمة تقريبية لـ π:

تم تنفيذ هذا المثال باستخدام الماتلاب:

إحدى طرق ايجاد قيمة تقريبية لـ π باستخدام مساحة ربع الدائرة ذات نصف القطر الواحدي الواقع ضمن مربع طول ضلعه يساوي 1 كما يوضح الشكل التالي:



عندئذ ولايجاد تقدير لقيمة نسبة مساحة ربع الدائرة إلى المربع نولد عدد من القيم العشوائية في المستوي والتي تتراوح قيمها بين 0,1، فإذا كان N_{tot} هو عدد النقط الكلي المولدة وكان N_{cir} عدد النقاط الواقعة ضمن ربع الدائرة عندئذ يكون: $\frac{A_{cir}}{A_{box}} \approx \frac{N_{cir}}{N_{tot}}$، لكن نعلم ان $A_{cir} = \frac{1}{4}[\pi R^2]_{R=1} = \frac{\pi}{4}$ وفيما يا يلي نص البرنامج باستخدام الماتلاب لتقدير قيمة π :

```
M = 5;% Repeate num
N = 1000;% Points
z = zeros(1,M);
data = zeros(N,M);
for j = 1:M,
    % Create Random points
    x = rand(1,N);
    y = rand(1,N);
    k = 0;% Stores Na
    for i = 1:N,
        if (x(i)^2 + y(i)^2) <= 1 % check if the
point is in pi-slice
            k = k + 1;%
        end
        data(i,j) = 4*k/i;
    end
    z(j) = data(N,j);
end
error = pi - data;
z %#ok<NOPTS>
mean(z)
var(z)
```

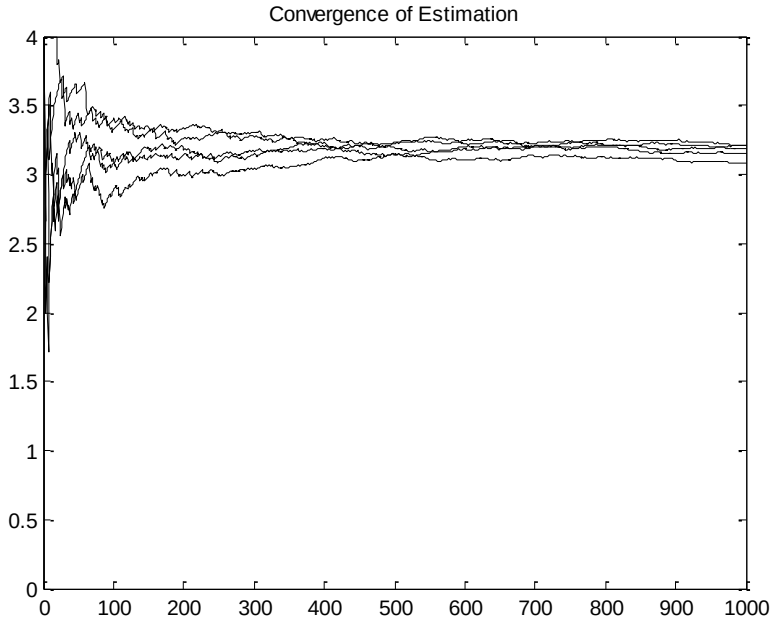
```
plot(data,'k'),title('Convergence of
Estimation');
figure, plot(error,'k'),title('Estimation
Error');
```

نتيجة تنفيذ هذا البرنامج يعطي التقديرات التالية:

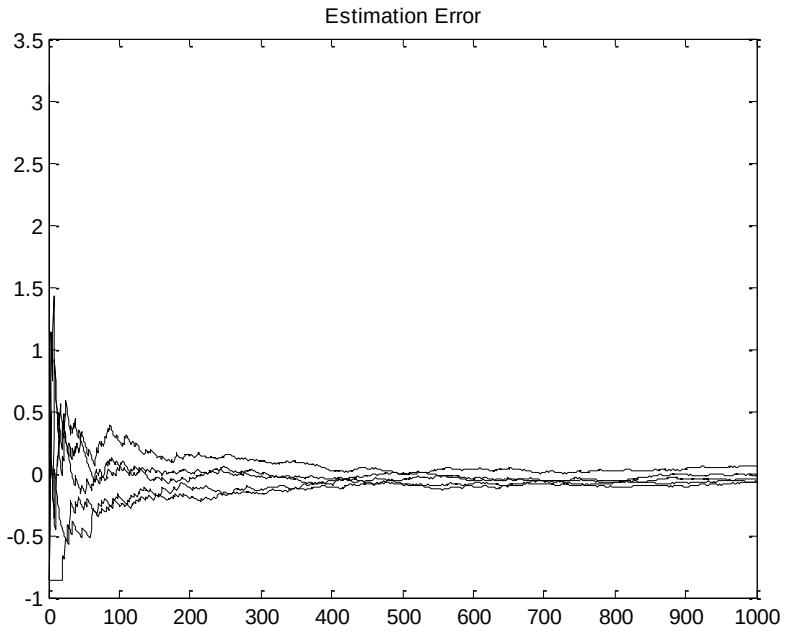
z = [3.1840 3.2120 3.1560 3.2080 3.0840]

mean(z) = 3.1688

var(z) = 0.0027



تقارب المقدرات.



خطأ التقدير

بأخذ المتوسط الحسابي لقيم المقدرات نحصل على القيمة: $\hat{\pi} = 3.1688$ وخطأ التقدير الكلي -0.0272.

يوضح المثال السابق طريقة إيجاد تقدير قيمة متحول ما باستخدام مقدرات مونت كارلو كما أنه يبين طريقة عمل محاكاة مونت كارلو وهي تتألف من الخطوات التالية:

- وجود شرط اختبار.
- وجود عدادين الأول تتم زيادته من أجل كل تجربة عشوائية والثاني عند تحقق الشرط الاختباري، فمثلا من أجل عمل محاكاة لنظام اتصالات ما والمعامل الذي يهمنا في هذه الحالة هو معدل الخطأ بالبت يكون الشرط الاختباري هو وجود خطأ في الاستقبال وهو الشرط الذي إذا تحقق نزيد من قيمة العداد الثاني في حين ان العداد الاول تتم زيادته من أجل كل بت تم ارساله.

قنوات الضجيج الابيض الغاوصي AWGN:

نريد تطبيق الخطوات السابقة في تحليل أداء نظام اتصالات ضمن قنوات الضجيج الأبيض الغاوصي الجمعي، ونقصد بالأداء معدل الخطأ بالبت، وبالتالي بالاعتماد على ما سبق فإنه لايجاد تقدير مونت كارلو لهذا المعامل (معدل الخطأ) يجب ارسال عدد من الرموز وليكن N،

وليكن عدد الأخطاء هو N_e عندئذ نقول ان تقدير معدل الخطأ (احتمال الخطأ بالبت) يعطى بالتقدير التالي:

$$\hat{P}_E = \frac{N_e}{N}$$

لكن هل هذا المُقدر غير منحاز؟ هل هو متماسك؟

تكون الاخطاء في هذه القنوات مستقلة عن بعضها، كما ان عدد الاخطاء N_e الذي نجده عند ارسال N رمز له توزيع كثير حدود *Binomial* لذلك يجب اولا دراسة توزع كثير الحدود.

توزع كثير الحدود:

نهدف من دراسة هذا التوزع إلى دراسة السلوك الاحصائي لـ $\hat{P}_E$ ونبدأ بحساب متوسط وتشتت N_e، من أجل أخطاء مستقلة عن بعضها يعطى احتمال ورود N_e خطأ عند ارسال N رمز بالعلاقة التالية:

$$pN(N_e) = \binom{N}{N_e} P_E^{N_e}(1-P_E)^{N-N_e}$$

حيث: $\binom{N}{k} = \frac{N!}{k!(N-k)!}$، P_E: احتمال حدوث خطأ وحيد.

بالتالي يكون المتوسط: $E\{N_e\} = NP_E$ والتشتت: $\sigma^2_{N_e} = NP_E(1-P_E)$ بالتعويض باستخدام مقدرات مونت كارلو نجد:

$$E\{\hat{P}_E\} = \frac{E\{N_e\}}{N} = \frac{NP_E}{N} = P_E$$

وبالتالي فإن هذا المقدر هو مقدر غير منزاح.

التشتت:

$$\sigma^2_{\hat{P}_E} = \frac{\sigma^2_{N_e}}{N^2} = \frac{P_E(1-P_E)}{N}$$

ومنه فإن: $\lim_{N\to\infty} \sigma^2_{\hat{P}_E} = 0$.

وبالتالي فإن المقدر لاحتمال الخطأ هو مقدر متماسك، نذكر أن الاستنتاج السابق صحيح في حال كون الاخطاء الواردة مستقلة عن بعضها.

نستنتج مما سبق انه لاجراء محاكاة لاداء نظام اتصالات فإنه يمكننا استخدام المُقدر:

$$\hat{P}_E = \frac{N_e}{N}$$

ونعلم انه بالمتوسط تعطينا محاكاة مونت كارلو النتيجة الحقيقية وذلك باحتمالية عالية (المقدر متماسك).

نذكر ان هذه النتائج التي حصلنا عليها صحيحة فقط في حال كون الاخطاء الواردة ضمن القناة هي أخطاء مستقلة عن بعضها، لكن ان كانت هذه الاخطاء مترابطة (كما في حالة القنوات ذات عرض المجال المحدود) فإنه لا يمكننا تطبيق المقدر السابق من حيث المبدأ، لكن يمكننا ذلك.

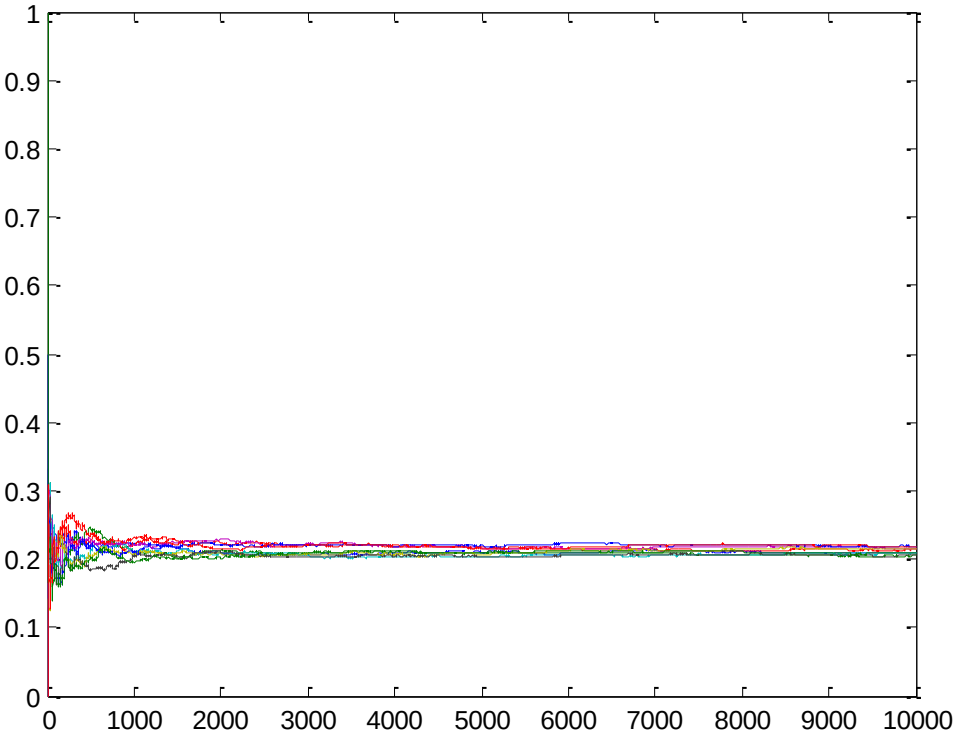
وفيما يلي نبين محاكاة لهذا المتحول في تقدير قيمة معدل الخطأ في نظام اتصالات غير مرمز والتعديل المستخدم *BPSK*:

```
M = 10;
N = 10000;
Data = randint(1,N,2);
bpsk_mod = modem.pskmod(2);
bpsk_dem = modem.pskdemod(2);

mod = modulate(bpsk_mod,Data);
snr = -5;
BER = zeros(N,M);
va = zeros(1,M);
for j = 1:M,
    mod1 = awgn(mod,snr);
    Ne = 0;
    dem1 = demodulate(bpsk_dem,mod1);
    for i=1:N,
        if dem1(i) ~= Data(i),
            Ne = Ne + 1;
        end
        BER(i,j) = Ne/i;
    end
    va(j) = var(BER(:,j));
```

```
end
plot(BER);
```

نتيجة التنفيذ نحصل على منحني تغيرات حساب معدل الخطأ من أجل كل بت جديد مستقبل ومن أجل كل تجربة على حدا وهذا ما يوضحه الشكل التالي:



الشكل 3: يبين تكرار تجربة قياس معدل الخطأ 10 مرات وفي كل مرة ارسال 10000 بت.

نلاحظ أنه مع زيادة عدد البتات فإن المقدر يسعى للاقتراب من قيمة وهي القيمة الحقيقية لمعدل الخطأ.

القيمة الوسطى للخطأ هي:

BER = 0.2127

التشتت في قيم خرج المقدر: $\sigma^2 = 4.6497e-010$

كما نعلم فإن لغات البرمجة في تطور مستمر وكل فترة زمنية تظهر لغة برمجة جديدة لها ميزات خاصة بها، وهذا يدفع المبرمجين إلى الانتقال لهذه اللغةالجديدة، فمثلاً لغات البرمجة C++, Visual Basic 6.0, Java كانت تعتبر من أقوى اللغات المستخدمة في تطوير البرامج ولكنها أصبحت قديمة مع ظهور شيء أفضل أو على الأقلى أحدث وهي البيئة .NET. التي قدتمها وتدعمها شركة مايكروسوفت في مجال هندسة البرمجيات Software Engineering.

قبل الخوض في تفاصيل هذه البيئة البرمجية الجديدة من الجيد أخذ نظرة عامة عن بيئات البرمجية السابقة.

- بيئة C/Win32 API: كان تطوير البرامج في بيئة ويندوز يعتمد بشكل رئيسي على لغة البرمجة C وترابطها مع توابع ويندوز Application Programming Interface (API). هذا الترابط أدى إلى تطوير مجموعة قوية من البرمجيات إلا أن تطوير البرامج بهذه الطريقة ليس سهلاً بل ويعتبر معقداً.

  كما نعلم فإن لغة C تزود المبرمج بالقدرة على التحكم بكل شيء وهي لا تدير أي شيء، فالمبرمج مسؤول عن حجز الذاكرة المطلوبة وإدارتها و تحريرها عند الانتهاء منها. تعتمد هذه اللغة بشكل رئيسي على السجلات Structures ولا تطبق قواعد البرمجة غرضية التوجه Object-Oriented.
- بيئة C++/MFC: تعتبر لغة C++ تطوير للغة C، ويمكننا النظر إليها على أنّها طبقة تدعم غرضية التوجه وتستند إلى الـ C، بهذا المفهوم نجد أن مبرمجي C++ يستفيدون من مفاهيم غرضية التوجه مثل: الوراثة Inheritance، التغليف Encapsulation، Polymorphism، ..... إلا أنهم

ما زالو يعانون من مشاكل الـ C المتمثلة في إدارة الذاكرة يدوياً، التعامل مع المؤشرات.

لكن وعلى الرغم من التعقيدات الموجودة في C++ فإن عدة بيئات موجودة اليوم تعتمد عليها، ونذكر منها صفوف مايكروسوفت التأسيسية Microsoft Foundation Classes (MFC). تزود هذه البيئة المبرمجين بالقدرة على تنفيذ توابع API المغلفة ضمن صفوف هذه البيئة. مما سبق فإن مهمة MFC تغليف توابع API.

- Visual Basic 6.0: نظراً للصعوبات التي عانى منها المبرمجون في عالم C++ فقد انتقلوا إلى عالم أكثر لطافة وأقل صعوبات حياتية وهو عالم Visual Basic. اشتُهرت هذه البيئة لأنها قادرة على بناء واجهات معقدة ،مكتبات كود Code Libraries مثل الـ COM ،التعامل مع المعطيات ،...

على الرغم من كون VB 6.0 تخفي تعقيدات API بشكل أفضل منه في MFC إلا أنها تعاني من مشكلة رئيسية وهي: أنها ليست غرضية التوجه بشكل كامل فمثلاً لا تسمح للمبرمج بإنشاء علاقة is-a المعروفة في غرضية التوجه. تم تجاوز هذه المشكلة في Visual Basic 6.0 .NET، والمشكلة الأخرى وهي مشكلة كبيرة أنّها لا تسمح بتعدد النياسق Multithreading.

- Java: تعتبر هذه اللغة من أوائل لغات البرمجة المبنية بشكل كامل تقريباً على مفاهي البرمجة غرضية التوجه وهي تشبه C++. إن قوة هذه اللغة ليس كونها مشتقة عن نظام التشغيل (كما نعلم) فحسب بل بكونها تزود المبرمجين بعدد كبير من الحزم المعرفة مسبقاً Predefined Packages التي تُمكن المبرمج من بناء تطبيق بشكل كامل بالاعتماد عليها فقط.

على الرغم من كون Java لغة جميلة إلا أنّ تطوير تطبيق يعتمد على Java يعني الاعتماد فقط عليها، كونها لا تدعم تعدد اللغات. لتوضيح هذه الفكرة

فإنّه في وقتنا الحاضر لا يعتمد التطبيق على لغة برمجة واحدة فقط، فمثلاً يكون قلب التطبيق Core يستخدم لغة C الواجهات تستخدم C# وهكذا.
كما أن تطبيقات الصور Graphic أو التطبيقات الحسابية لا يُفضل استخدام Java فيها، وذلك لأن هذه اللغة تحتاج إلى وجود طبيقة برمجية بين التطبيق ونظام التشغيل، هذه الطبقة هي الجهاز الوهمي Virtual Machine (JVM).

هذه كانت أشهر البيئات وأكثرها انتشاراً بين المبرمجين المبتدئين، طبعاً توجد بيئات غيرها مثل Component Object Model (COM) ، Windows DNA Programming، .....

كما رأينا فإن المبرمجين بحاجة إلى بيئة تطوير جديدة، هذه البيئة كانت .NET

الحل .NET:

كما رأينا فإن الحياة كمبرمج ويندوز صعبة للغاية، فجاءت بيئة التطوير .NET لتجعل حياة المبرمج أسهل.

الحل الذي طرحته هذه البيئة هو تغيير كل شيء، كما سنرى في مواضيع لاحقة فإن هذه البيئة جديدة في مجال تطوير تطبيقات لنظام ويندوز وغيره.

في ما يلي بعض محاسن هذه البيئة:

- إمكانية استخدام الكودات الموجودة حالياً Interoperability: وهذه ميزة جدية جداً، لأنّه في وقتنا الحاضر هناك عدد كبير من مكتبات COM ،مكتبات C ، وغيرها. وبالتالي نستطيع استخدامها جميعاً دون مشاكل.
- التكامل الكامل والتام مع لغات البرمجة الأخرى: هذه البيئة تدعم جميع عمليات غرضية التوجه بين لغات مختلفة.

- مكتبة صف أساسي Base Class Library: هذه المكتبة تحتوي على الصف الأساسي الذي تُشتق منه جميع الصفوف، وتخفي هذه المكتبة تعقيدات API.
- بيئة تطوير سهلة: فتطوير تطبيق يعتمد على .NET لا يحتاج إلى تسجيل لبنى ثنائية جديدة في مسجل الويندوز Registry مثل تطوير تطبيقات تعتمد DirectX التي تحتاج عمليات تسجيل.

Abstract

Optics have greatly developed over the past decades, making it usable in wired communication systems, it has many advantages such as:

- High channel capacity.
- Interference immunity.
- Low loss.

That’s why it became so popular in different types of usage.

This thesis discuses the security of a fiber optic and proposes an intrusion detection system able to detect different intrusion techniques and specify the intrusion point.

المراجع

[1]. *Max Brown and Emil Wolf* **, " Principles of Optics"**, *7th edition.*

[2]. *Jose T.Vigil, "* **An Evaluation of Fiber Optic Intrusion Detection Systems in Interior Applications".** *Sandia National Laboratories.*

[3]. " Fiber Optic Intrusion Detection Systems". *Network Integrity.*

[4]. *R.hui, M.O'Sullivan,* **" Fiber Optics Measurement Techniques".**

[5]. *D.R.Anderson, L.Johnson, F.G.Bell,* **" Troubleshooting Optical Fiber networks".**

[6]. *Ajoy Ghatak, "* **Introduction to Fiber Optics",** *Cambridge.*

[7]. *JIM HAYES*, **" Fiber Optics Technician's Manual".**

[8]. *Marcell Dakkar, "* **Practical Fiber Optics",** *2002.*

[9]. *Harry J. R. Dutton,* **" Optical Comm Tutorial"** *, 1998 .*

[10]. *J.H.FRANZ V.K.JAIN,* **" Optical Communications Components and Systems".**

[11]. *D.F. Walls, G.J.Milburn* **, " Quantum Optics".**

[12]. *Francis T.S YU, Shizhou Yin, "* **Fiber Optics Sensors".**

[13]. *David Belly, Edwin Wright, Newnes, 2003.*

[14]. " 8100 Modules series ", User Manual.

[15]. *TOM G.Brown,* **" HANDBOOK OF OPTICS".**

[16]. *Oyster Optics, Inc.* **"Securing Fiber Optic Communications against Optical Tapping Methods"**