MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention

Du Diplôme de Master II en Electronique

Option : Réseaux et télécommunication

Thème :
Conception et réalisation d'un réseau Vsat

Proposé et dirigé par :
Mr. M. AZOUNI
Mr. H. SEGHIR

Présenté par :
Mr. HADDALAH Mohamed
Mr. SADAT Mohamed

Co promoteur:
Mr. M. LAHDIR

Promotion 2010-2011

SOMMAIRE

CHAPITRE II: LES SRVICES VSAT

CHAPITRE III. ETUDE DE LA LIAISON VSAT

INTRODUCTION GENERALE

De nos jours, nous remarquons que les satellites prennent de plus en plus de place dans les télécommunications et leur fonction a changé avec les décennies. Parallèlement au déclin relatif de leur rôle dans la téléphonie, ils ont pris une importante croissance dans la télédiffusion et dans la diffusion de données. Ils amorcent un retour médiatisé dans la téléphonie.

Les satellites sont des éléments indispensables d'une toile spatiale qui recouvre de plus en plus le globe terrestre et qui aujourd'hui, permet des communications de n'importe quel point de la planète.

La technologie de transmission bidirectionnelle existe depuis les années 1980, elle est connue sous le nom de Vsat (Verry Small Aperture Terminal). Au tout début, ces réseaux Vsat étaient réservés à des environnements professionnels spécifiques tels que les plates-formes pétrolières, les camps d'expédition, les bases polaires .etc. puis les Vsat se sont généralisés au niveau des entreprises qui souhaitaient déployer des réseaux dans les zones délaissées ou inaccessibles par les opérateurs. Les tarifs des équipements et des services Vsat pour les entreprises restent relativement élevés et ne sont donc pas accessibles aux particuliers.

L'installation d'une station Vsat nécessite une étude préalable afin de rentabiliser au maximum le système une fois installé. L'entreprise dispose des différents guides d'installation spécifiques à chaque équipement. Cependant la création d'une procédure générale d'installation d'une station terrestre, après l'étude de la technologie Vsat faite dans ce rapport, peut aider un technicien désirant intégrer la société ou un stagiaire à mieux comprendre et d'acquérir les bases nécessaires pour bien accomplir sa tâche.

Dans cette étude, nous allons présenter les concepts et les éléments clés pour la compréhension des technologies satellites et plus particulièrement la technologie Vsat afin de déterminer les spécificités de ce type de technologie de transmission.

Notre étude comportera trois axes principaux. Le premier chapitre consacré pour l'étude de la technologie Vsat. Nous commencerons par rappeler quelques notions sur les satellites, puis nous analyserons de manière détaillée la structure et le fonctionnement d'une

station terrestre Vsat. Dans le deuxième chapitre, nous allons nous intéresser aux services Vsat.

Quant au troisième chapitre, nous allons aborder le calcul du bilan de liaison lors d'une communication entre deux stations au sol et un satellite, en introduisant les composants d'un système de communication par satellite et les paramètres qu'il faudrait spécifier lors de la mise en place. Et enfin, nous allons terminer par une conclusion générale.

CHAPITRE I.

ETUDE DE LA TECHNOLOGIE VSAT

I.1. Introduction :

La technologie VSAT (Very Small Aperture Terminal) est apparue il y a une vingtaine d'années aux Etats Unis d'Amérique. Au fur et à mesure, le système a été amélioré et sa démocratisation a permis de faire baisser les prix des matériels. Aujourd'hui, certains opérateurs et fournisseur d'accès on fait l'acquisition de hubs et louent des accès à des entreprises qui n'ont pas les moyens financiers suffisants pour posséder leur propre hub. Ainsi, de petites entreprises peuvent d'interconnecter plusieurs points très distants entre eux pour un coût équivalent ou inferieur à un système filaire. Certains fournisseurs d'accès proposent également des accès Internet pour les particuliers avec des débits et des tarifs équivalents à des systèmes filaires.

I.2. Présentation de l'entreprise :

L'opérateur Wataniya Télécom Algérie (WTA), le premier opérateur multimédia de téléphonie mobile en Algérie a obtenu une licence de desserte nationale des services de téléphonie sans fil en Algérie le 2 décembre 2003.

Le 25 août 2004, Wataniya a procédé au lancement commercial de sa marque Nedjma, assorti de services et d'avantages encore jamais égalés dans le pays. Nedjma introduit de nouveaux standards dans l'industrie des télécommunications en Algérie.

Le réseau Nedjma a été déployé pour offrir aux consommateurs algériens des communications de qualité exceptionnelle en émission et en réception.

Nedjma utilise le réseau GSM sur les fréquences 900/1800.

I.2.1. Qualité de service :

Nedjma offre aux utilisateurs algériens un nouveau monde en matière de télécommunications mobiles. En effet, Nedjma met au service de la clientèle algérienne non seulement des produits, mais aussi une haute qualité de transmission grâce à des équipements issus des technologies les plus récentes, un service à la clientèle basé sur les standards les plus élevés.

I.2.2. cahier de charge :

Les premières semaines passées à Nedjma étaient consacrées à l'étude théorique des systèmes de télécommunications par satellites ce qui nous a donné la chance de comprendre pas mal de choses pour un domaine qu'il fallait bien maitriser en tant que Master en réseau et télécommunication. Pour cela, plusieurs documents que contient l'entreprise ont été mis à notre disposition.

Après l'étude théorique des systèmes Vsat, nous avons passées à la réalisation d'un logiciel sous matlab avec l'aide de notre tuteur pour calculer tous les paramètres d'une liaison satellite et c'est l'objectif de ce stage qui à duré 6 mois.

I.3. Notion sur les satellites :

Avant de s'intéresser a la technologie Vsat, il est bon de rappeler quelques notions sur les satellites.

I.3.1.Historique :

L'idée de placer un objet en orbite autour de la terre à commencé à germer au début de ce siècle. En fait, c'est le britannique Arthur.c.clorcke qui introduit en 1945 le concept de communication par satellite.

En 1957, les russes ont lancés leur premier satellite appelé sputnik-1 doté d'un émetteur radio suivi par sputnik-2 en 1958. Pendant ce temps, les américains expérimentaient les répéteurs passifs et actifs.

Les répéteurs passifs, leur rôle est de réfléchir des signaux émis par les stations terrestres. Pour ce mode, ‘’ECHO ‘’ (ce satellite était un ballon en plastique aluminé de 30 mètres de diamètre) est le premier satellite lancé en 1960.

Les répéteurs actifs. Ces satellites possèdent leur propre système de réception et d'émission capable d'amplifier le signal et de transmettre une plus grande quantité d'informations.

Malgré toutes ces découvertes, il fallait attendre 1965 pour voir le satellite rentre dans le domaine public et signe le début de l'ère des télécommunications par satellite telle que nous la connaissons aujourd'hui.

I.3.2. Définition :

Un satellite est une sorte de relais hertzien. Il est considérer comme un élément indispensable d'une toile spatiale qui recouvre de plus en plus le globe terrestre et qui aujourd'hui, permet des communications de n'importe quel point de la planète. Son rôle est de régénérer le signal qu'il a reçu et de le retransmettre amplifié en fréquence vers plusieurs stations réceptrices qui se trouvent sur la surface de la terre et inversement.

I.3.3. Structure du satellite :

Dans la structure d'un satellite, on trouve deux parties principales : la plate-forme et la charge utile.

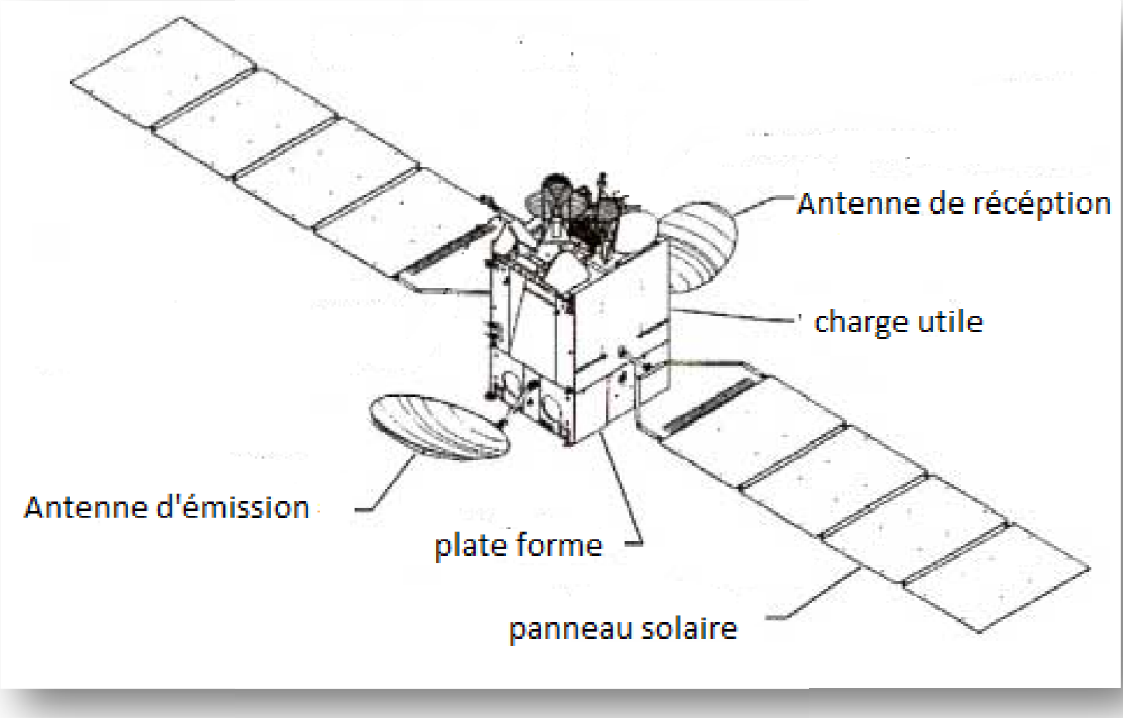


Figure I.1 : structure d'un satellite

La plate-forme comporte l'ensemble des sous-systèmes permettant à la charge utile de fonctionner. Elle comprend :

- **L'alimentation électrique:**

  Pour fonctionner, les satellites ont besoin d'énergie fournie par le soleil. Ce système d'alimentation en énergie utilise des panneaux solaires pour convertir la lumière en énergie électrique, des batteries pour la stocker, et un système de distribution qui transmet l'énergie électrique à chaque instrument.

- **Le système de commande :**

  C'est le cerveau du satellite. Il permet de contrôler toutes les fonctions du satellite.

- **Le contrôle d'altitude et d'orbite et les équipements de propulsion :**

  Ce système permet au satellite de rester stable et toujours orienté dans la bonne direction. Le satellite possède des capteurs qui lui permettent de connaitre son orientation. De plus, le satellite a aussi besoin de pouvoir se déplacer pour corriger sa position, c'est pourquoi il possède un mécanisme de propulsion. La performance du système de contrôle d'altitude dépend de l'utilisation du satellite. Un satellite utilisé pour faire des observations scientifiques a besoin d'un système de contrôle d'une plus grande précision que pour un satellite de télécommunication.

- **Les équipements de poursuite, de télémesure et de télécommande TT&C :**

  Ces équipements se composent d'un système émetteur/récepteur, ainsi que de diverses antennes permettant de relayer les informations entre la terre et le satellite. La base de contrôle au sol utilise ces équipements pour transmettre de nouvelles instructions à l'ordinateur du satellite. Ce système permet aussi de transmettre des images ou autres formes de données enregistrées, aux ingénieurs se trouvant sur terre.

- **Le contrôle thermique :**

  Le système protège tous les équipements du satellite des dommages dus à l'environnement spatial. En orbite, un satellite est exposé à des températures (allant de -120° lorsque le satellite est dans le noir, jusqu'à 180° lorsque le satellite se trouve exposé au soleil). Le contrôle de la température utilise une unité de distribution de chaleur et un système de couverture thermique pour

protéger les équipements électroniques du satellite de ces brusques changements de température.

La charge utile d'un satellite représente tous les équipements qui permettent au satellite de réaliser la fonction pour laquelle il est destiné. Elle est composée de caméras digitales et de capteurs d'image pour prendre des clichés de la surface de la terre.

I.3.4. Orbite des satellites :

Un satellite tourne autour du globe grâce aux lois de la gravitation, décrivant une trajectoire en forme d'ellipse ou de cercle dont le plan passe par le centre de la terre. Sa vitesse étant inversement proportionnelle à son altitude, elle est donc minimale lorsque le satellite est à l'apogée de son orbite (point de la trajectoire le plus éloigné de la terre) et maximale lorsqu'il se trouve à son périgée (point de la trajectoire le plus proche de la terre) comme nous montre la figure suivante.

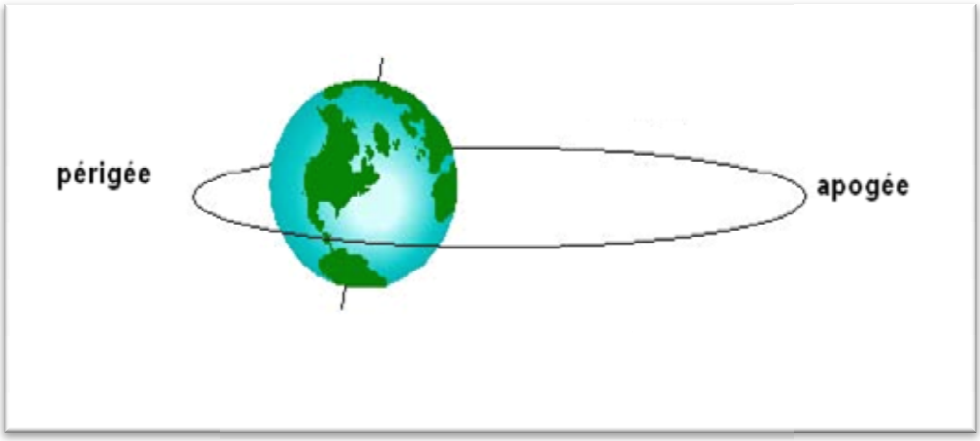


Figure I.2 : orbite des satellites

On distingue 3 types d'orbites

I.3.4.1. Orbite géostationnaire :

Les satellites GEO (Geostationary Earth Orbit) sont les plus communs grâce à leur position relative fixe dans l'espace et ce malgré leur coût important, un délai important et une diffusion importante non adaptée à certaines applications.

Placés au-dessus de l'équateur à 35 786 km d'altitude, ces satellites géostationnaires effectuent leur révolution en 23 heurs 56 minutes et 4 secondes, durée qui correspondant à la période de rotation de la terre. Se déplaçant dans le même sens et à la même vitesse que le

globe, ils apparaissent ainsi immobiles depuis le sol et peuvent couvrir instantanément une large calotte équivalente environ à un hémisphère.

Comme les satellites géostationnaires conservent toujours la même position par rapport à la terre, ils peuvent donc être associés à des antennes terrestres fixes. Ils présentent cependant l'inconvénient d'être situés bas sur l'horizon lorsqu'ils couvrent des zones éloignées de l'équateur : les signaux à transmettre, parcourant une plus grande distance, subissent des atténuations plus importantes et mettent plus de temps à arriver sur Terre.

Les différentes particularités de l'orbite géostationnaire :

Comme on à dit précédemment, le satellite géostationnaire a la particularité de rester fixe par rapport à un observateur terrestre. Pour se faire, l'orbite aura plusieurs caractéristiques :

- Orbite circulaire : le foyer de l'orbite est le centre de la terre. Le satellite et toujours à la même vitesse et à la même distance par rapport à la terre. Dans ce cas, la notion de périgée et d'apogée n'existe plus.
- L'orbite équatoriale : un satellite en orbite équatoriale gravite directement au-dessus de l'équateur : son inclinaison est nulle.
- Orbite synchrone : le satellite est à 35786 Km d'altitude. Sa période orbitale représente la période de rotation de la terre, c'est-à-dire 23heurs 56 minutes et 4 secondes.

I.3.4.2. Orbite LEO (Low Earth Orbit):

Les orbites basses (LEO) entre 200 et 800 km. Le satellite reste visible au-dessus d'un point, que quelques minutes. Pour sa capture, il nécessite des antennes suiveuses de dimensions convenables. Ces orbites sont utilisées par les navettes, les laboratoires spatiaux, l'observation et la photographie de la terre, la météorologie, ainsi que les satellites militaires.

I.3.4.3. Orbite MEO (Medium Earth Ordit):

Les orbites moyennes (MEO) entre 10 000 et 15 000 km. Le satellite reste visible au-dessus d'un point pendant quelques heures. Leurs utilisations sont du même ordre que pour les satellites en orbite basse (LEO).

I.4. La transmission du signal satellite :

I.4.1. Fréquence utilisée :

Pour éviter un chaos total dans le ciel, une réglementation internationale spécifique et stricte a été mise en place par l'Union Internationale des Télécommunications (UIT-T) concernant la répartition des fréquences ; elle fait partie intégrante du règlement international des radios communications. Cette réglementation définit notamment la position orbitale des satellites et les bandes de fréquences qu'ils doivent utiliser et respecter.

Les systèmes vsat utilisent des bandes de fréquences particulières. Les plus communes sont les bandes C, Ku, et la bande Ka.

La bande C : destinée au trafic commercial par satellite. Cette bande dispose deux plages de fréquence, le sens montant (terre-satellite) est compris entre 5.9 et 6.4 GHz, et le sens descendant entre 3.7 et 4.2 GHz.

La bande Ku : le sens montant est compris entre 14 et 14.5 GHz, le sens descendant entre 10.7 et 11.7 GHz. Cette bande est employée pour la radiodiffusion par satellite, elle est aussi utilisée pour les systèmes d'accès internet par satellite.

La bande Ka : elle peut être utilisée pour les voies remontantes des connexions internet par satellite. L'avantage de la bande Ka et sa fréquence plus élevée (20 à 40 GHz) que la bande Ku : les antennes peuvent être plus petites pour un même débit. Donc la bande Ka et réellement un facteur de baisse des coûts pour l'internet par satellite.

I.4.2. les signaux radioélectriques :

Pour être transmis par satellite, les signaux délivrés à la station terrienne modulent une porteuse radioélectrique. Cette porteuse est reçue par le satellite, lors d'une liaison montante, et ensuite la station terrienne destinatrice reçoit également à son tour la porteuse, lors de la liaison descendante.

I.4.3. Type de modulation :

La transmission satellite utilise les technologies analogiques et numériques. En analogique, on exploite la modulation FM et en numérique les modulations de phase PSK (Phase Shift Keying).

La modulation FM et très résistante aux non linéarités des amplificateurs et permet d'accroitre le rapport signal sur bruit grâce à l'augmentation du taux de modulation sans

accroissement de puissance. Cette propriété se fait au détriment de la bande passante qui augmente elle-aussi.

La modulation de phase elle également très résistante aux non linéarités. La 2PSK à été la première utilisée pour sa simplicité de réalisation, puis remplacée par la QPSK (Quadrature Phase Shift Keying) qui aujourd'hui exploitée à 99%.

I.4.4. Le codage :

Le codage est employé pour transmettre des médias sensibles aux erreurs, tels que la voix et la vidéo compressées et les données. C'est la meilleure méthode pour réduire la probabilité d'erreur en ajoutant de bite de redondance à l'émission. Le récepteur utilise ces bits de redondances pour détecter et corriger les bits d'informations éventuellement erronés. Le taux de codage est exprimé à l'aide d'un rapport FEC (Forward Error Correction) qui traduit le nombre de bits d'informations utile par rapport au nombre de bits transmis. Les principales techniques de codage sont :

- Le codage par blocs introduise une redondance qui dépend uniquement des éléments du message à transmettre. Parmi ces codes on trouve : les codes simples et le code de Hamming.
- Le codage convolutionel, où les bits transmis résultent d'une combinaison entre les bits d'informations et les bits de redondance.

I.4.5. Les politiques d'accès aux canaux satellites :

Si deux stations émettent en même temps, il y aura collision des signaux qui deviennent incompréhensibles puisque impossibles à décoder, donc Il y aura perte des deux messages qui devront être retransmis ultérieurement.

La politique d'accès aux canaux satellites a été réalisée pour permettre à plusieurs stations ou utilisateurs d'accéder à un même canal de transmission, et pouvoir avoir une exploitation maximale des transpondeurs du satellite tout en garantissant qu'il y ait le moins de collisions possibles.

Les trois principales politiques d'accès qu'on peut trouver dans le domaine des communications par satellite sont :

- Les politiques de réservation (AMRF, AMRT, AMRC)
-Les politiques d'accès aléatoire (ALOHA……..)
-Les politiques de réservation par paquet.

a. Les politiques de réservation :

(i) Accès Multiple à Répartition en Fréquence (AMRF ou FDMA)

C'est une technique analogique. Elle s'agit d'un découpage en bande de fréquences de manière à attribuer une partie du spectre à chaque station et si de nouvelles stations se joignent au système de communication, il est obligatoire d'assigner de nouvelles bandes de fréquences.

Chaque station doit doter : d'un modulateur, un émetteur, de n récepteurs, et de n démodulateurs pour pouvoir mettre en place cette technique.

(ii) Accès Multiple à Répartition dans le temps (AMRT ou TDMA)

C'est la technique la plus utilisée actuellement dans le domaine des transmissions par satellites. Son principe est de découper le temps entre les différentes stations. Par cette technique, une fréquence peut être utilisée par plusieurs stations tout en utilisant la totalité de la bande passante.

(iii) Accès Multiple par répartition en code (AMRC ou CDMA)

Le principe de cette méthode est l'allocation de canal par durée et non par paquet. Ceci en utilisant un identifiant chacune de stations du système de communication. Cette technique permet à plusieurs stations équipées chacune de leur propre code numérique d'utiliser simultanément la même onde porteuse sans interférer les unes sur les autres. En effet, toutes les stations vont émettre sur le même canal en même temps, avec la même fréquence, et chacune de ces stations pourra reconnaître les données qui lui sont destinées grâce au code d'identification approprié.

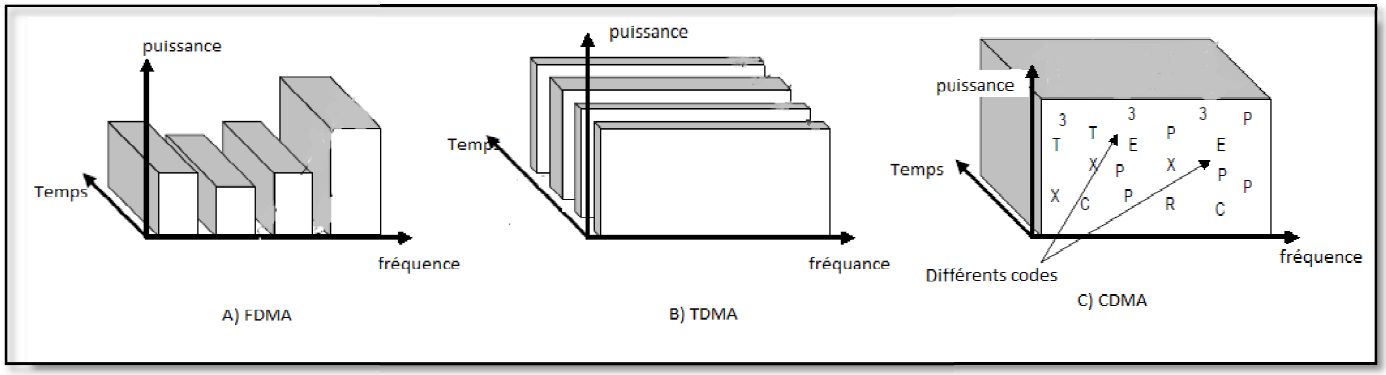


Figure I.3 : les politiques de réservation

b. Les politiques d'accès aléatoire

Les politiques d'accès aléatoire pour les réseaux satellites sont les mêmes que celles définies pour les réseaux locaux. Nous allons en définir quelques unes successivement parmi : la technique ALOHA, ALOHA en tranche et ALOHA avec réservation.

(i) La technique ALOHA

Dans cette technique, les données sont transmises de façon sourde c'est-à-dire que la station émet ses données sans chercher à savoir si le support est déjà occupé. Le problème majeur posé par cette technique, est que plusieurs stations peuvent émettre en même temps, ce qui mène à des interférences entre les différentes émissions, et à l'altération des données. Si ce problème aura lieu, la station va retransmettre les paquets après un délai aléatoire.

(ii) ALOHA en tranches ou discrétisé

L'idée de cette technique est de découper le temps en tranches correspondant chacune au temps de transmission d'un paquet. Son avantage est que, s'il y a détection de collision, c'est sur l'ensemble de la tranche de temps, et non sur une partie d'un paquet. Cette méthode de découpage du temps en tranches a donnée une amélioration pour le taux d'utilisation du canal.

(iii) ALOHA avec réservation

Pour cette technique, si une station commence à émettre un paquet, il y a de fortes chances qu'elle en émettre un autre en même temps. Donc il faudra réserver plusieurs tranches de temps à une station qui commence à émettre. Lors de la présence de collision, celle-ci s'effectue sur un intervalle complet et non sur une partie.

c. Les politiques de réservation par paquet

Le but de ces méthodes est de permettre, aux stations qui en ont vraiment besoin, de transmettre leurs données.

Les politiques de réservation par paquet sont très nombreuses. Le dénominateur commun de ces méthodes réside dans la faculté de réserver à l'avance des tranches de temps pour les stations qui ont des paquets à émettre.

Parmi ces méthodes, on trouve :

-Réservation par une file d'attente fictive, premier arrivé, premier servi.

-Réservation ordonnée : Cette méthode utilise une trame avec autant de mini tranches dans l'en-tête de réservation qu'il y a de tranches. Les minis tranches sont dédiées aux stations et leurs permettent de prévenir les autres stations de l'utilisation ou non de sa tranche de temps dans la prochaine trame. Dans le cas où une mini tranche n'est pas utilisée par la station correspondante, elle devient libre d'accès et les autres stations peuvent y accéder en mode aléatoire pour réserver une tranche de temps.

Le choix de ces techniques d'accès aux canaux satellites dépend essentiellement :

-de la qualité d'information à transmettre.

-du nombre de stations à gérer.

I.5. Technologie vsat :

Le système Vsat est un standard d'accès bidirectionnel basé sur des satellites géostationnaires permettant l'émission et la réception de données à partir d'un terminal de petite dimension variant de 1 à 3.7 mètres de diamètre. Il est implanté dans des pays qui ont une très grande superficie et dans lequel le réseau filaire est peu développé. C'est un service utilisé particulièrement par des entreprises pour communiquer librement et en permanence avec tous ses bureaux ou filiales.

L'avantage de ce système est que, il peut gérer, au niveau réseau des applications haut débit pouvant atteindre des vitesses de transmission de 8Mbps en voie descendante et ≤2Mbps en voie montante par rapport aux nombres d'utilisateurs.

Son principal inconvénient est son prix très élevé. C'est pour cette raison que seules les grandes entreprises peuvent investir de sommes importantes pour leur infrastructure réseau.

I.5.1. Structure d'un réseau Vsat :

Comme la montre la figure I.4, un système de communication par satellite est constitué : d'un secteur spatial composé par un satellite et une station de contrôle située au sol appelée ‘’hub’’ et d'un secteur terrestre composé d'une antenne parabolique et des équipements reliés au réseau filaire.

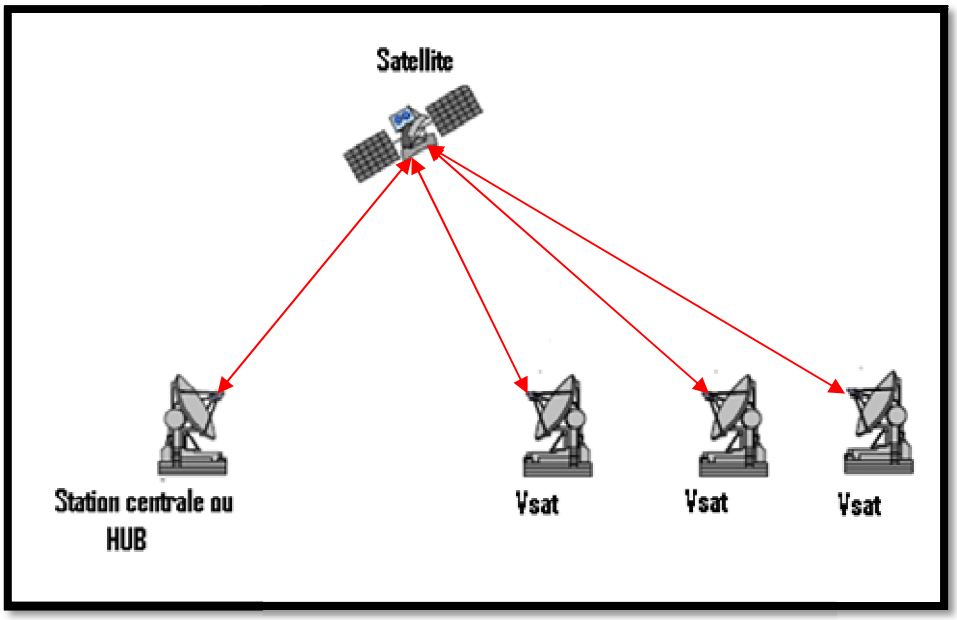


Figure II.4 : structure d'un réseau Vsat

I.5.1.1. Station spatiale :

En plus de satellite (situé souvent sur une orbite géostationnaire) dont le rôle est d'assurer le relais, l'amplification, et la transposition du signal transmis. Le secteur spatial à deux autres équipements importants qui sont : le transpondeur et la station hub.

- **Les transpondeurs :**

Un satellite transporte un certain nombre de répéteurs appelés transpondeurs, matériels qui reçoivent des signaux de la liaison montante affaiblissent par la distance qu'ils ont parcourus, les démodulent, puis les remodulent à la fréquence de liaison descendante et les amplifient avant de les réémettre vers la terre.

Les transpondeurs sont identifiés par leurs fréquences montantes et descendantes.

- **La station de contrôle Hub :**

Le Hub est le point le plus important du réseau. Indépendamment de la taille et du nombre de sous-ensembles, il y'à peu de différence entre lui et un Vsat. De part son importance, sa structure est conséquente : une antenne entre 5 et 7 mètres de diamètre. C'est par lui que transite toutes les données qui circulent sur le réseau et c'est aussi lui qui gère tous les accès à la bande passante. Des bases de données produites par le Hub sont également employées pour la facturation.

La station Hub est équipée d'un système de gestion de réseau NMS (Network Management System) relié à chaque station Vsat à l'aide des circuits virtuels permanents.

Le NMS permet à un opérateur de réseau pour surveiller et contrôler le statut du Hub et de chaque station Vsat, et en cas d'interruption dans les stations Vsat, il télécharge tous les logiciels et les paramètres du système pour le relancement de l'opération.

I.5.1.2. Station terrienne Vsat :

La figure I.5 nous donne l'architecture d'une station Vsat. Comme la montre cette figure, la station Vsat peut divisée en deux ensembles d'équipements reliés par une paire de câbles : l'unité extérieure ODU (Outdoor unit) avec l'antenne parabolique et l'unité intérieure IDU (Indoor unit).

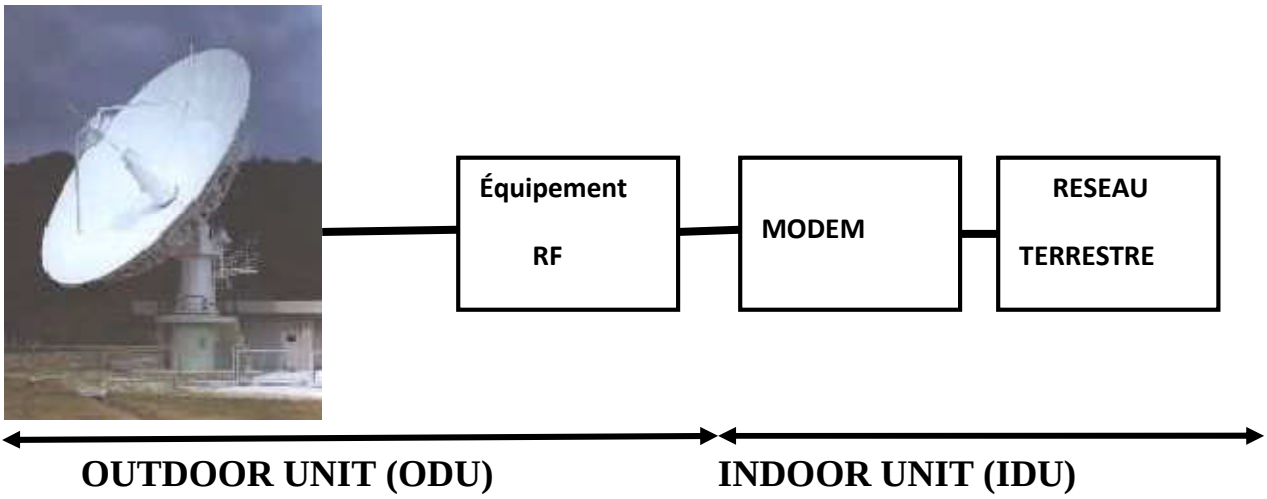


Figure I.5: configuration typique d'une station terrienne

a. Outdoor Unit (ODU) :

Un ODU est l'équipement situé à l'extérieure d'un bâtiment, y compris l'antenne parabolique, et un équipement RF d'ou on trouve : convertisseur à faible bruit (LNB) et un bloc_up-converter (BUC) dont le rôle est d'amplifier la transmission de liaison montante.

(i) Parabole :

La parabole ou antenne à réflecteur parabolique est un élément incontournable de la réception satellite. C'est elle qui capte le signal venant du satellite et qui le concentre vers le convertisseur. Comme c'est illustré dans la figure I.6, La parabole est constituée de deux éléments distincts : la source (feed), et le réflecteur conducteur qui influe directement sur les caractéristiques de rayonnement de l'antenne.

Figure I.6 : La parabole

(ii) La source (feed) :

Le cornet d'alimentation, situé au point focal du réflecteur parabolique, qui rayonne l'énergie RF vers le réflecteur d'antenne ou recueille l'énergie RF reçue du réflecteur d'antenne et redirige un signal vers les réflecteurs de l'antenne qui le transmettent vers le satellite.

- ***La polarisation :***

Le signal provenant du BUC entre directement dans l'OMT (OrthoMode Transducer) ou s'effectue la polarisation. La polarisation du signal est effectuée pour séparer les fréquences d'émission et celles de réception. En général, en polarisation linéaire, l'émission se fait en polarisation horizontale et la réception en verticale. La polarisation circulaire est aussi utilisée.

(iii) Equipement RF :

L'équipement Radiofréquence a deux fonctions essentielles dans la transmission par satellite, le but initial étant de minimiser les pertes lors de la transmission. Dès lors, cette partie du système va se charger d'une part, d'amplifier le signal pour pourvoir être traité par les équipements, et d'autre part, de convertir les fréquences de travail du satellite en fréquences intermédiaires et vice-versa.

- ***LNB (Low Noise Block) :***

En réception satellite, le signal recueilli par l'antenne parabolique est trop faible pour pouvoir l'exploiter directement : il faut l'amplifier, c'est l'un des rôles du LNB nommé

également tête de réception. C'est convertisseur de signaux intégrés dans les antennes paraboliques permettant la réception analogique et numérique de signaux satellites.

Le rôle du LNB est capital. Il recueille le signal de 12GHz reçu par la parabole puis le convertit en une fréquence intermédiaire BIS (Bande Intermédiaire Satellite) comprise entre 1 et 2GHz. A cette fréquence les atténuations dans le câble de liaison sont moins importantes et le traitement des signaux par le récepteur s'en trouve facilité.

- ***BUC (Block Up Converter) :***

Un Block Up Converter est un système utilisé pour la transmission des liaisons montantes des signaux satellites. Il convertit une bande de fréquence, d'une fréquence basse vers une fréquence plus élevée pour pouvoir être correctement traitée par le transpondeur spatial avec moins d'interférences possible.

Le BUC s'interface avec la parabole en bande C ou en bande Ku, permettant à un modem d'émettre sur la liaison montante vers un satellite.

Sur les équipements Vsat, les BUC sont généralement employés en même temps que les LNB. Le BUC, étant un dispositif d'émission de la liaison montante, il s'accorde (pour transmettre) avec l'OMT vers le feed.

Le côté réception du système s'accorde aussi sur l'OMT, lorsque le LNB est le dispositif du convertisseur bas de la liaison descendante. Le LNB s'accorde (pour recevoir) alors du coté réception.

b. Indoor Unit (IDU) :

Cette partie est reliée à l'ODU par un simple câble (distance maximale : environ 60m). Le rôle de l'IDU est de transformer à l'aide d'un Modem le signal reçu à partir de l'antenne parabolique afin qu'il soit exploitable par un ordinateur.

(i) Le Modem :

Le Modem se charge de transformer les données arrivées en entrée. Cet appareil peut recevoir un signal modulé en hautes fréquences et le transformer en informations basses fréquences. En réception satellite, il permet l'obtention des signaux audio, vidéo et données véhiculées par une onde porteuse, afin de les restituer via les circuits d'un ordinateur. Il transforme les fréquences en tensions et traite l'information de façon à ce qu'elle soit lue par un ordinateur.

En ce qui concerne les modulations utilisées par les Modem satellites, on trouve les modulations de phase (BPSK, QPSK, 8-PSK etc.) choisies en fonction de la bande passante.

Nous y trouvons aussi un multiplexeur connecté au modem chargé de concaténer les données de types varies provenant du réseau filaire dans des trames.

I.5.2. Architecture du réseau Vsat :

Les réseaux Vsat se présentent sous deux principales architectures de communication étoile ou maillée), très fiables et faciles à installer.

I.5.2.1. La topologie en étoile :

Dans l'architecture en étoile représentée par la figure I.7 , le hub joue le rôle de station terrienne centrale qui contrôle et communique avec un grand nombre de stations utilisateurs. Dans ce cas, les stations ne peuvent pas communiquer s'ils ne passent pas par le hub.

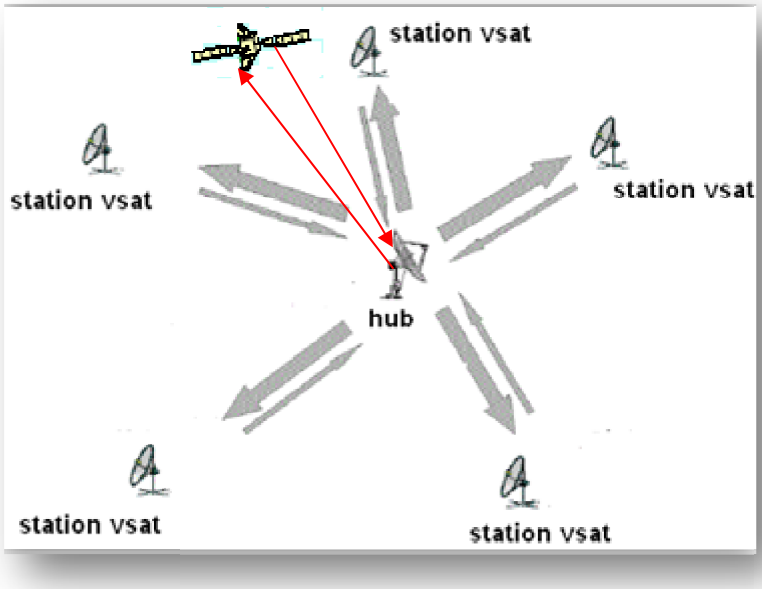


Figure I.7 : topologie en étoile

I.5.2.2. La topologie maillée:

La topologie maillée permet eux différentes stations distantes de communiquer entre eux sans passer par le concentrateur (hub) ce type de connexion réduit les retards et ses applications sont nomreuses : la téléphonie, le réseau d'entreprises…..etc.

Le schéma suivant nous montre comment une liaison de type maillée s'éffectue (figure I.8).

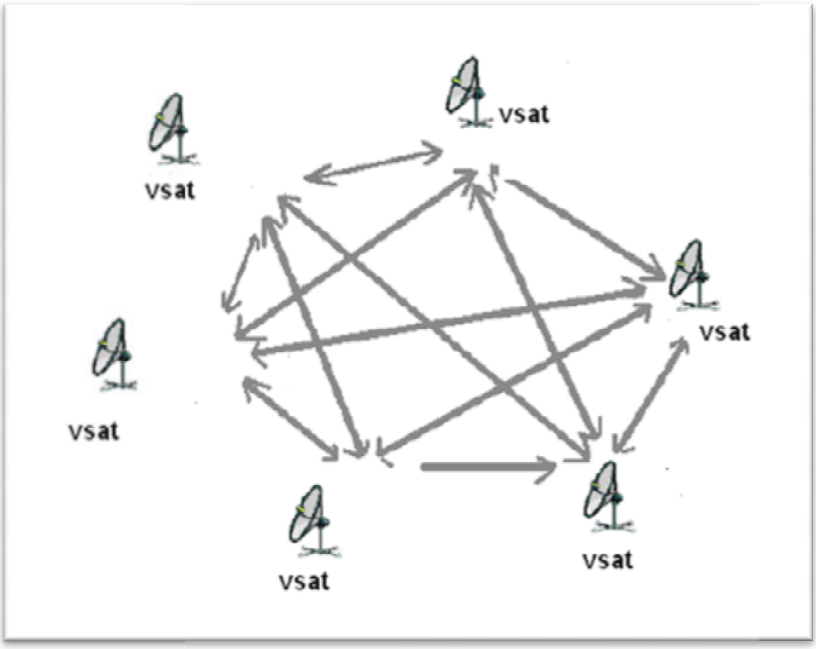


Figure I.8 : topologie maillée

I.6. Réseau Vsat WTA

Le réseau Vsat wta est constitué d'un Hub, de 23 stations Abis et de 07 stations Aters, les deux topologies point à point et point à multi point sont utilisés sur un spectre de 24 MHZ qui est exploité à 98% pour couvrir tous les BSC du Sud ainsi les sites lointains dans le sud Algérien et le réseau est toujours en extensions pour l'intégration de nouveaux sites Abis et Aters.

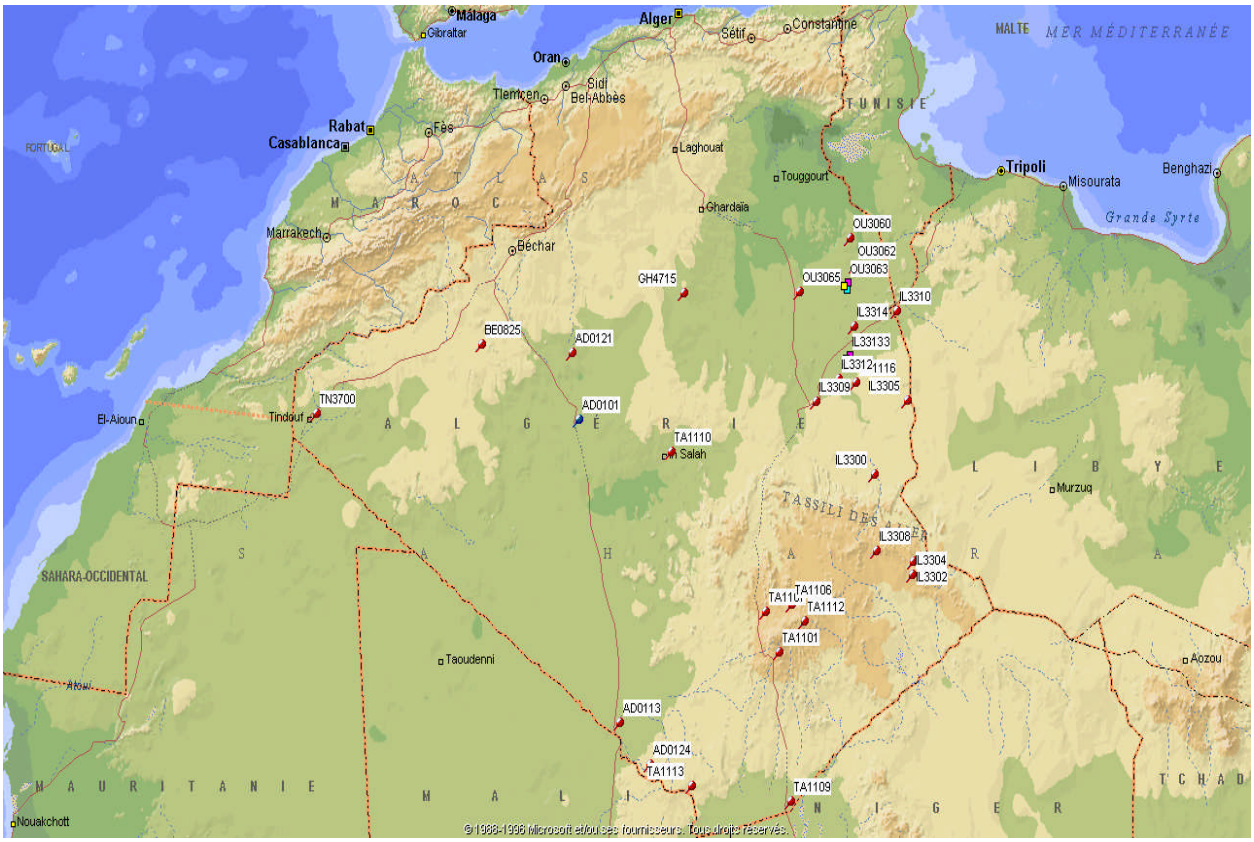


Figure I.9 : Cartes sites Vsat WTA

I.6.1. Gains avec les réseaux Vsat actuel :

- une meilleure stabilité, sécurité et disponibilité pour les BSC du Sud : vu que le Vsat est plus stable que les liaisons fibre Optique qui nécessitent beaucoup de répéteurs et de points de coupures.
- couverture des endroits lointains dans le sud (y compris les Bases de pétrole)
- résiliation des liaisons LL (liaison louée ou leased line : Liaison de télécommunication établie entre deux points d'un réseau public et réservée à un client pendant une durée convenue dans le cadre d'un contrat de location) et qui coute généralement très cher selon la distance entre les deux points qu'on veut relier.
- Gain en charge et dépenses.
- Amortissement du prix des équipements Vsat et spectre au bout d'une Année et gain et bénéfice assuré.

I.6.2. Exemple d'une architecture Vsat WTA :

On utilise deux équipements essentiels pour notre transmission Vsat :

- **Modem :** pour la transmission et réception du signal qui porte le trafic et les données voulus
- **CXU** : équipement d'optimisation du débit Vsat et qui permet d'avoir une optimisation de 40% sur le spectre, afin d'optimiser au maximum la bande passante qui est relativement cher.

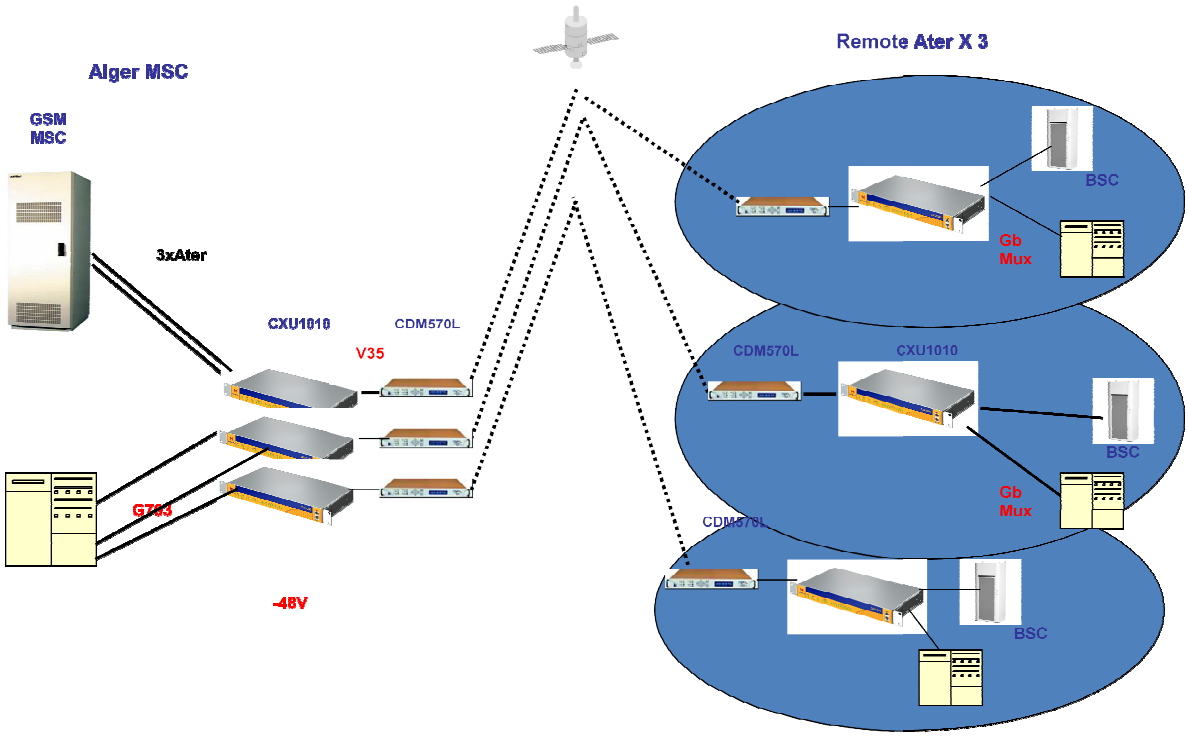


Figure I.10 : Architecture du réseau Vsat Wataniya

I.6.3. Projets futures :

-Extension de la bande passante du spectre, achat d'un nouveau spectre

-introduction des nouveaux équipements pour plus d'optimisation sur le spectre (c'est des modems à haute technologie et qui permettent d'optimiser le spectre à environ de 50%)

- nouvelles intégrations Abis et Aters, pour plus de couverture, plus de disponibilité et stabilité pour les sites Sud.

I.7. Conclusion :

Dans ce chapitre, nous avons présenté une étude sur.la technologie VSAT portant sur la composition de ce système et ses liaisons, Nous avons ensuite passé en revue le réseau VSAT de l'entreprise WATANIA et ses projets futurs.

CHAPITRE II.
LES SERVICES VSAT

II.1. Introduction :

Vsat est un système qui est prévu pour mettre en place des réseaux de données, mais depuis son apparition dans les années 80, des améliorations ont été apportées au système et les constructeurs ont réussi à augmenter considérablement le nombre d'applications possible avec un tel réseau.

Aujourd'hui, le réseau Vsat n'est plus seulement un réseau de données, mais il peut nous offre d'autres améliorations pour les réseaux téléphoniques et pour les systèmes de localisation qui offrent des services majoritaires dans le domaine de télécommunication actuel ainsi pour les prochaines années.

Dans cette partie, nous attarderons sur les systèmes de localisation par satellite, leur principe de fonctionnement et les nouvelles utilisations. Puis, nous allons parler de la transmission des données via satellite. Ensuite nous allons aborder le domaine de la téléphonie mobile par satellite.

II.2. Le GPS (Global Position System) :

Pendant les temps, les seuls aides à la navigation ont été les étoiles, le compas, la sonde, les cartes ou les gyroscopes. Puis des émetteurs radio terrestres ont permis aux marins et aux aviateurs de déterminer précisément leur position, mais ces systèmes ne couvraient pas l'ensemble du globe. Grâce aux satellites, aujourd'hui, avoir la position exacte d'un objet sur toute la surface de la terre est rendu possible.

Le GPS est un système de positionnement par satellites, nous permet de connaître la position d'un objet a n'import quel point du globe. Il comprend au moins 24 satellites en orbite autour de la terre, à une altitude de plus de 20000 km sur six plans orbitaux, émettent en permanence des signaux qui peuvent être captés de n'import quel point du globe.

Le département américain de la défense à commencé la mise en place de ce système de localisation au milieu des années 70, afin de guider ses navires, ses avions et ses véhicules

terrestres, n'importe où dans le monde. Puis les civils ont eu accès au réseau, mais avec une précision réduite.

En mesurant le temps de propagation des signaux émis par ou moins quatre satellites, les récepteurs GPS calculent la position à 18 mètres près, voire seulement à 1 mètre quand ils reçoivent également les signaux d'un émetteur terrestre du voisinage.

II.2.1. Principe de fonctionnement :

Le GPS fonctionne grâce au calcul de la distance qui sépare un récepteur GPS et plusieurs satellites. Cette distance est effectuée grâce à des signaux radio émis par le satellite : des impulsions radio sont émises à des instants connus, de sorte que, en mesurant les temps d'arrivée des signaux, un récepteur calcule les instances qui les séparent du satellite. On définit ainsi des sphères centrées sur des satellites et dont l'intersection donne la position du terminal. Ce dernier est capable d'identifier le satellite qu'il utilise à l'aide du signal pseudo aléatoire émis par chaque satellite. Il charge à l'aide de ce signal, les informations sur l'orbite et la position du satellite.

Pour trouver la position exacte d'un équipement, il faut qu'il se trouve dans la zone de couverture d'au moins trois satellites. Plus le nombre de satellites est important, plus la position donnée est correcte.

II.2.2. Les applications GPS :

Ce système est largement utilisé par les navigateurs, les géomètres et les aviateurs. Les applications sont nombreuses : des véhicules équipés de récepteurs GPS calculent des itinéraires, des trajets de transport, plus rapides, qui évitent les embouteillages par exemple, et permettant ainsi de gagner du temps. Il permet aussi aux avions de voler plus directement vers leur destination sans être gênés par l'encombrement des couloirs aériens, et d'atterrir en douceur sans aucune visibilité.

II.3. la téléphonie mobile GSM:

Nous connaissons tous très bien les téléphones mobiles du réseau GSM, et nous savons aussi que ce réseau a le défaut de ne pas couvrir tout le territoire, car 80% de la planète restera à jamais non couverte par les réseaux de téléphones cellulaires.

Depuis quelques années, il se développe de nouveaux réseaux satellite. Le trafic vocal assuré par les satellites, et surtout destiné aux pays qui n'ont pas de réseaux en fibres

optiques. Avant de s'intéresser à l'accès du réseau téléphonique mobile par la liaison satellite, il est bon de rappeler quelques notions sur le réseau GSM.

II.3.1. Définition :

Le GSM (Global System for Mobile Communications), première norme de téléphonie cellulaire de seconde génération qui soit pleinement numérique, fournit des services de transmission de la voix et éventuellement de données à un débit de 9.6kbps. L'architecture du réseau GSM repose sur un ensemble d'équipements spécifiques aux réseaux mobiles. La topologie du réseau GSM est simplifiée pour se concentrer uniquement sur les éléments dont on aura besoin (voir figure II.2).

II.3.2. la cellule et la station de base :

Dans un réseau GSM, le territoire est découpé en petites zones appelées cellules.
Chaque cellule est équipée d'une station de base fixe munie de ses antennes installées sur point haut (château d'eau, immeuble….).

Les cellules sont dessinées hexagonales mais la portée réelle des stations dépend de la configuration du territoire arrosé et du diagramme de rayonnement des antennes d'émission.

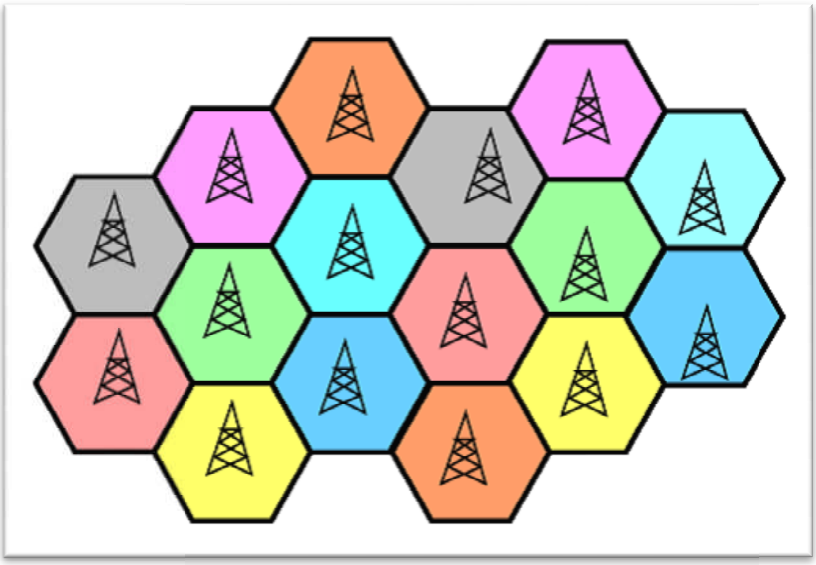


Figure II.1 : La division d'une région couverte en Cellules

Dans une cellule GSM typique (macro cellule), les mobiles peuvent être situés jusqu'à 35 km de la station de base pour le GSM900 et 2 km (mini cellule) pour le DCS1800 (puissance plus faible, atténuation plus importante avec la distance).
La taille limitée des cellules permet de limiter la puissance d'émission nécessaire pour la liaison et donc augmenter l'autonomie des mobiles.

II.3.3. Architecture du réseau GSM :

Le réseau GSM est composé de trois sous-ensembles :

a. Le Sous-système Radio BSS (Base Station Sub-system) :

Le sous système radio correspond à la fonction de distribution du réseau de radiocommunication. Il permet les transmissions radioélectriques et gère la ressource radio. Il est constitué d'une ou plusieurs stations de base BTS et d'un contrôleur de station BSC.

- ***BTS (Base Transceiver Station) :***

C'est un ensemble d'émetteurs/ récepteurs appelés TRX. Son rôle est la gestion des transmissions radios (modulation, démodulation, égalisation, codage et correcteur d'erreurs). Chaque BTS réalise la couverture radio d'un certain territoire (appelé « cellule ») dont le rayon varie entre quelques centaine de mètres et quelques kilomètres.

- ***BSC (Base Station Controller) :***

Permet de contrôler un ensemble de BTS et possède son registre d'abonnés visiteurs VLR stockant les informations sur les abonnés présents dans les cellules gérées. Il prend en charge l'allocation de la ressource radio et la décision de l'activation/désactivation d'un canal vers un mobile.

b. Le Sous-système d'acheminement-NSS (Network Sub System) :

Il comprend l'ensemble des fonctions nécessaires à l'établissement des appels. Il est composé des éléments suivants :

- ***MSC (Mobil Switching Center) :***

Il s'agit essentiellement d'un commutateur qui constitue le nœud central du réseau de téléphonie mobile. Il comporte des équipements informatiques qui gèrent l'acheminement des informations à travers le réseau GSM. C'est également le MSC qui permet de connaitre à tout moment la localisation d'un téléphone mobile dans le réseau.

- ***VLR (Visitors Location Register) :***

C'est une base de données associée à chaque MSC. Le VLR contient une partie des informations des HLR concernant les abonnées des mobiles situés dans les BSS dépendant du MSC. Le VLR enregistre les informations de localisation des mobiles.

- ***HLR (Home Location Register) :***

Contient les informations nécessaires à la gestion des communications d'un certain nombre d'abonnés. Pour chaque abonné qu'il gère, le HLR possède l'identité internationale de l'abonné (IMSI), et son numéro MSISDN.

c. Le Sous-système d'exploitation et de maintenance-OSS :

Permet à l'exploitant d'administrer le réseau (coûts, performances, erreurs, sécurités……..).

- ***AUC (Authentification center) :***

Assure l'authentification des terminaux du réseau.

- ***EIR (Equipement Identity register) :***

Une base de données annexe contenant les identités des terminaux.

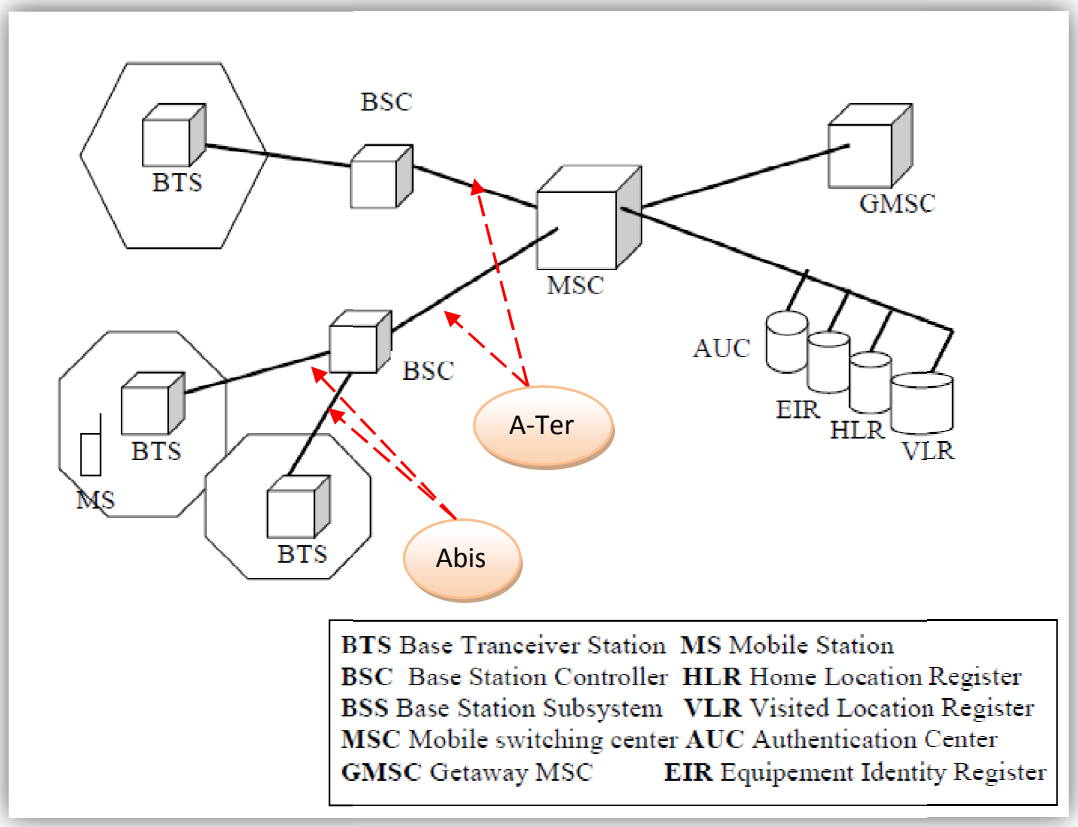


Figure II.2 : Structure du réseau GSM simplifié

II.3.4. Présentation des interfaces et l'interface Abis :

Nom de l'interface	Localisation	Utilisation
Um	MS-BTS	Interface radio
Abis	BTS-BSC	Divers
A	BSC-MSC	Divers
C	GMSC-HLR	Interrogation du HLR pour message court entrant
	SM- GMSC-HLR	Interrogation du HLR pour message court entrant.
D	VLR-HLR	Gestion des informations d'abonnés et de localisation
	VLR-HLR	Services supplémentaires
E	MSC- MS- GMSC	Transport de messages courts
	MSC-MSC	Exécution des handover
G	VLR-VLR	Gestion des informations des abonnées
F	MSC-EIR	Vérification de l'identité de terminal
B	MSC-VLR	Divers
H	HLR-AUC	Echange des données d'authentification

Tableau II.1 : description des interfaces

Dans le contexte de la communication téléphonie mobile via satellite, chaque interface peut être employée comme liaison satellite. Là, on s'intéresse à l'interface Abis.

II.3.4.1. Définition :

Abis est une interface définie entre la BTS et le BSC. Abis supporte la transmission des communications des usagers et de la signalisation. En réalité, la plupart des messages de signalisation sont échangés entre le BSC ou le MSC et la MS (Mobil Station) : la BTS n'a qu'une simple fonction de relais.

L'avantage que présente l'interface Abis est que, la voix transmise sur cette interface peut utiliser un format compressé, c'est-à-dire que capacité nécessaire pour un appel est de 16Kbps par rapport à celle de l'interface A (située entre le BSC et le MSC) qui est de 64Kbps. Ce débit faible est plus adapté au contexte de communication par satellite où la bande passante est relativement contrainte. C'est la raison pour laquelle l'interface Abis peut être adoptée comme un point de coupure pour l'introduction de La topologie GSM par satellite comme c'est illustré sur la figure II.3.

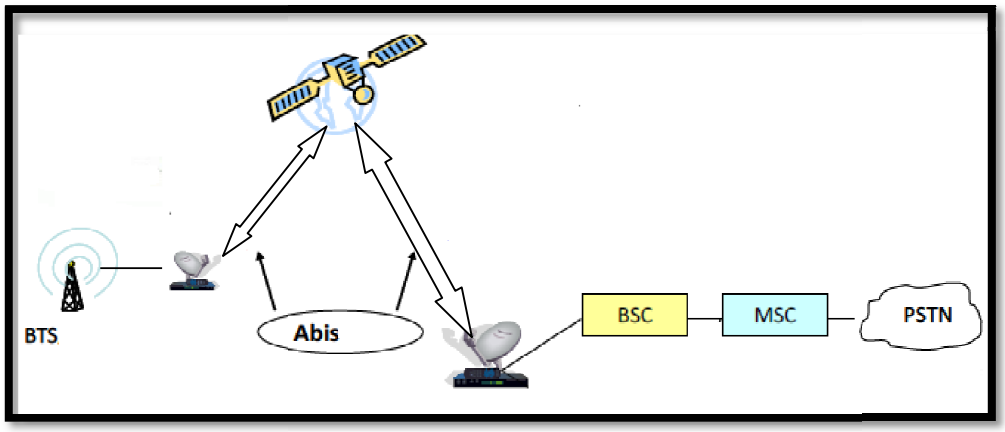


Figure II.3 : Topologie GSM par satellite

II.3.5. Les fréquences de travail :

Dans le système de téléphonie mobile GSM900 et DCS1800 (Digital Cellular System (ou GSM1800)), deux bandes de fréquences sont utilisées, l'une autour des 900Mhz et l'autre autour de 1.8 GHz. Dans le cas du réseau GSM900, la bande de fréquence comprise entre 880 et 915 MHz est utilisée pour la transmission du téléphone mobile vers l'antenne-relais (voie montante), tandis que la bande comprise entre 925 et 960 MHz est utilisée pour la transmission de l'antenne-relais vers le mobile (voie descendante).comme la montre la figure suivante :

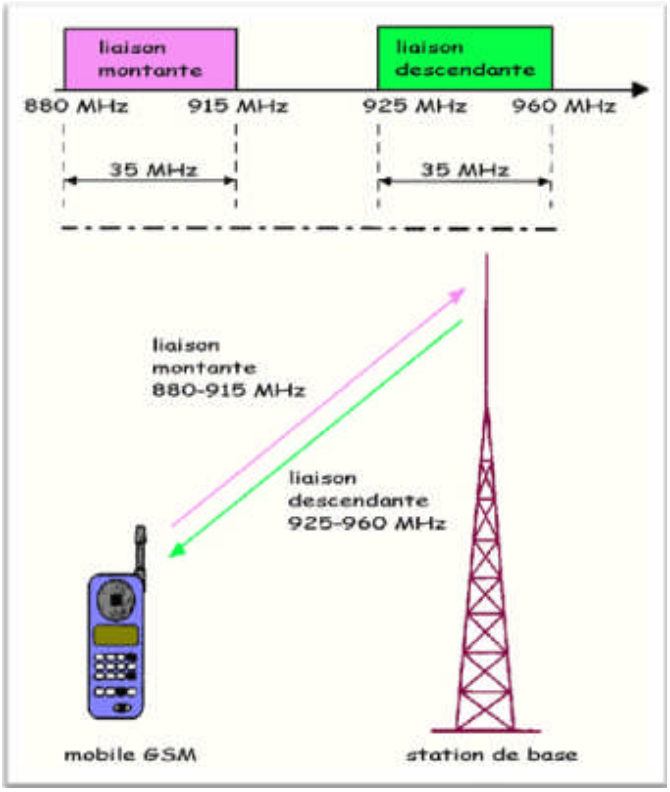


Figure II.4. Liaison entre mobile et station de base de GSM

II.3.6. Les Objectifs du GSM dans la télécommunication :

- ❖ Offrir un vaste éventail de services de télécommunications compatibles avec ceux des réseaux fixes.
- ❖ Offrir des services spécifiques dus à la mobilité des usagers.
- ❖ Assurer la compatibilité d'accès à n'importe quel utilisateur dans n'importe quel pays exploitant le système GSM.
- ❖ Assurer la localisation automatique des mobiles sous la couverture globale de l'ensemble des réseaux.
- ❖ Permettre une grande variété de terminaux mobiles.
- ❖ Obtenir une bonne efficacité spectrale.
- ❖ Obtenir des coûts permettant d'assurer le succès du service.

II.3.7. Connexion de réseau GSM :

Alors que les réseaux de communications s'étendent à des régions toujours plus isolées, l'infrastructure terrestre requise pour interconnecter les réseaux GSM n'est pas toujours suffisants. Les satellites apportent une solution idéale de raccordement des réseaux GSM pour élargir la couverture cellulaire, créer une dorsale de réseau ou encore fournir une solution de secours en cas de problème sur le réseau terrestre.

Les satellites sont aux cœurs même de ces réseaux, assurant la connectivité entre les stations de bases distantes d'émission-réception (BTS) et les contrôleurs de station (BSC), ou

entre les contrôleurs et les centres de commutation mobiles (MSC). En fait, les satellites transmettent simultanément des milliers de conversations téléphoniques. Les appels via des satellites dirigés vers la station terrestre la plus proche. La séquence de bits codant ces appels module une onde qui amplifiée et transmise au satellite par une antenne parabolique.

Arrivée au satellite, cette onde est très faible, elle doit être amplifiée avant d'être retransmise à la station réceptrice sur la terre, où les signaux individuels de paroles sont reconstitués.

Les techniques de compression éliminent le silence redondant et les blocs libres à la fois sur les liens ABIS (BSC vers BTS) et A-TER (MSC vers BSC), optimisant l'usage de la bande passante du satellite et réduisant les frais de transport sur le réseau GSM.

Le raccordement GSM par satellite vous évite de passer par l'infrastructure terrestre et participe au désenclavement des régions les plus isolées.

Le satellite fourni également des services privés aux entreprises et gouvernement signataires, pour leur usage propre comme la communication Vsat au sein d'une entreprise.

II.4. La transmission de données (l'Internet par satellite) :

Pour diffuser des chaînes de télévision ou fournir des services Internet, les opérateurs satellite utilisent des satellites géostationnaires. La principale force du satellite réside dans le fait que le déploiement au sol est immédiat. En effet, contrairement à l'ADSL ou au câble, l'Internet par satellite ne nécessite aucun réseau terrestre.

Le système repose simplement sur :

- Le centre opérationnel terrestre (téléport) qui centralise toutes les données émises et reçues par les différents satellites de la flotte.
- Le satellite ou plutôt les réseaux de satellites situés dans l'espace constituant la flotte de l'opérateur.
- Une parabole à installer directement sur le lieu de réception (domicile ou entreprise).

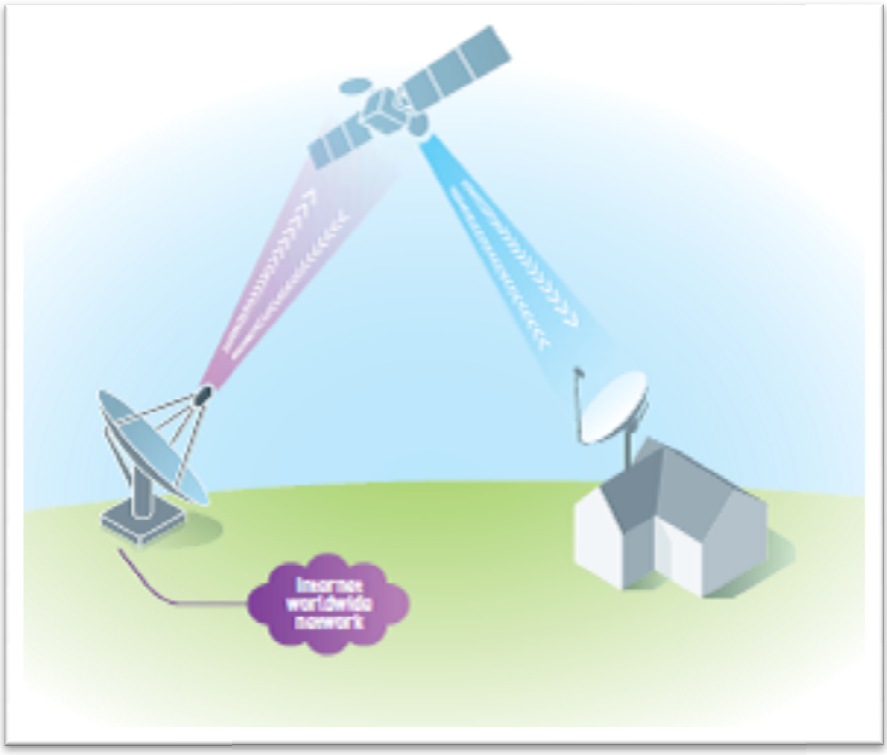


Figure II.5 : Principe de fonctionnement de l'internet par satellite

II.5. Conclusion :

Dans ce chapitre, nous nous sommes focalisés sur l'étude des services via satellite et nous avons constaté que les applications sont très nombreuses : téléphonique, transmission de données, localisation……..

Grâce aux satellites, la possibilité d'êtres joignable n'importe où sur terre est rendue possible.

CHAPITRE III.

ETUDE DE LA LIAISON VSAT

III.1. Introduction :

L'objectif de cette partie est le détail du calcul du bilan de liaison Vsat pour pouvoir effectuer la transmission avec la qualité requise. C'est donc une partie essentiellement mathématique qui envisage la liaison dans son ensemble en calculant tous les paramètres nécessaires, commençant par le bilan de fréquence passant par les paramètres d'antenne et enfin le bilan de puissance.

III.2. Bilan de fréquence :

Le modem travaille à une fréquence intermédiaire L-Band (950-1450MHz). Pour transmettre et recevoir aux fréquences (C-Band ou KU-Band), il d'abord régler les fréquences de travail de l'indoor unit. Le calcul des fréquences se fait de la manière suivante :

- **En bande RF :**
  - *Fréquence reçue* : $Fr = \frac{r\,début + r\,in}{}$ (GHz)……(1)
  - *Fréquence transmise* : Ft=Fr+ 2800……..……..……(2)

Avec :

$Fr_{début}$ est la fréquence au début du spectre donnée en GHz.

Fr_{fin} est la fréquence à la fin du spectre donnée en GHz.

2800 Hz est la fréquence au niveau du satellite.

- *Bande de fréquence Allouée :*

$$BPA = \frac{}{(\ *\ \ *\ \ )} * \ .$$ ……..……(3)

Avec :

1.35 est l'efficacité spectrale de la bande passante : c'est pour assurer un espacement entre les spectres et éviter les chevauchements.

n : est le facteur de modulation :*BPSK=1, QPSK=2, 8PSK=3, 8QAM=3, 16QAM=4.*

D : est le débit (Kb/s).

FEC : est le facteur de correction d'erreur= $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, \frac{7}{8}$

FEC$_{RS}$: est le Reed Sdomon≅ 1.

Les Forward Error Correction (FEC) détectent l'erreur et ils ont la possibilité de la corriger sans nécessiter la réémission de l'information comme dans le cas des codes de détection qui font uniquement la détection.

- *Bande passante Occupée :*

$$BPO = \frac{é}{.} \quad \text{……..…..(4)}$$

- **En bande L :**

Remarque :

Au niveau du *Hub : on utilise un BUC 80 Watts avec une fréquence de modulation de 13050 MHz.*

Au niveau de la station : *on utilise un BUC 4 watts avec une fréquence de modulation de 12800 MHz.*

Le calcul des fréquences se fait de la manière suivante :

- ***Pour la transmission :***
  - Au niveau de la station : Fréquence TX$_L$=TX$_{RF}$-12800…….(5)
  - Au niveau du HUB : Fréquence TX$_L$=TX$_{RF}$-13050…….(6)
- ***Pour la réception :***
  - Fréquence RX$_L$=RX$_{RF}$-10000.

Avec, TX$_{RF}$ est la fréquence d'émission RF de l'antenne, et RX$_{RF}$ est la fréquence de réception RF de l'antenne.

III.3. Paramètres d'antenne :

Dans cette partie, nous donnons les résultats principaux de la théorie des antennes :les grandeurs fondamentales, gain et ouverture. Mais avant, il est bien de rappeler les types d'antennes paraboliques et les types de montures.

III.3.1. Types de parabole :

Comme c'est illustré à la figure III.1, il existe quatre types de paraboles qui présentent des avantages et des contraintes.

- L'antenne prime focus est la plus classique. Imposée pour des diamètres supérieurs à 1 mètre, son cornet est au foyer primaire du réflecteur.il à un rendement de 60% et d'une température de bruit élevée.
- L'antenne Offset est le plus répondue pour les petits diamètres (<1m). son cornet est décalé dans le foyer secondaire afin de l'orienter différemment. Cette technique nous permet donc d'obtenir un rendement de 70 à 80% avec une température de bruit faible.
- L'antenne Cassegrain utilisée pour les grands diamètres (>3m). elle possède un réflecteur parabolique principal et un réflecteur secondaire de type hyperboïde utilisée pour réfléchir les ondes venant du réflecteur primaire vers la source. Dans ce type de montage la source est placée au centre du réflecteur avec une réduction de la température de bruit.
- Antenne grégorienne moins courante. Elle est utilisée pour les transmissions militaires ou la réception TV. Elle combine les avantages de l'antenne Offset et Cassegrain.

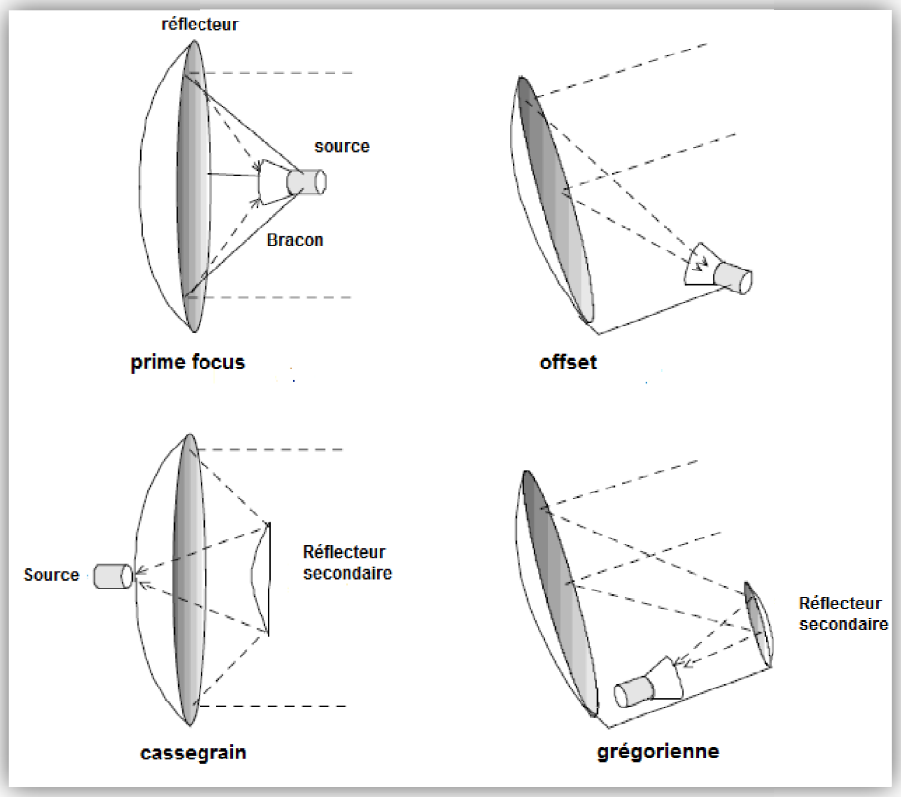


Figure III.1 : type de parabole.

III.3.2. Type de montures :

Le rôle de monture est d'orienter et suivre dans les meilleures conditions possibles un satellite. La monture est choisie selon le mouvement que soit rapide, lent, sur une orbite GEO ou à défilement. On distingue trois types de montures comme le montre la figure III.2 :

- Monture XY.
- Monture azel (azimut_ élévation) contient un axe vertical (azimut) et un axe horizontal (élévation).
- Monture polaire très adaptée pour les satellites géostationnaires. Permet de pointer tous les satellites visibles au sol.son mouvement suit un axe parallèle à l'axe de pôles terrestres.

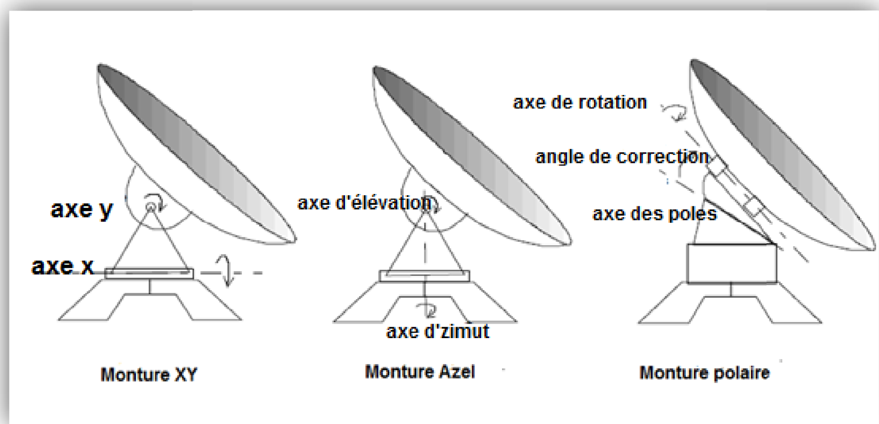


Figure III.2 : type de monture.

III.3.3. Calcul des paramètres d'antenne parabolique :

a. Longueur d'onde émise : en (mètre)

$$\lambda = -$$ [m]…………………………….(1)

On à comme données :

C : vitesse de la lumière qui est égale à 300 MHz.

f: fréquence d'onde.

b. La largeur du faisceau a -3dB :

La formule de La largeur du faisceau d'une antenne est donnée comme suite :

$$\theta = K\frac{\lambda}{}$$ [radian]……………………….(2)

K : coefficient entre 50 et 70. (ici, on le prend 70) (exprimé en %).

D : Diamètre d'antenne parabolique [m].

λ: la longueur d'onde.

On aura enfin notre formule suivante :

$$\theta = \frac{}{} \quad \text{[radian]} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

f : fréquence en GHz.

c. Distance de focalisation :

Comme c'est illustré a la figure III.3, les paramètres mathématique d'une antenne parabolique sont sa distance focale et son diamètre.

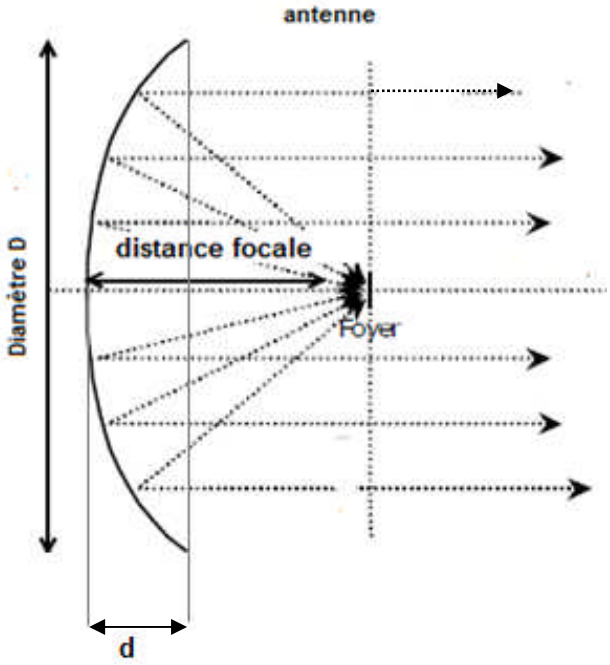


Figure III.3 : Distance de focalisation.

La distance de focalisation se calcule est donnée à la formule suivante :

$$f/d = \frac{}{} \quad \text{[m]} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

D : diamètre d'antenne.

d : profondeur d'antenne.

d. Distance station terrienne-satellite :

Comme nous montre la figure suivante, les lieux terrestres sont repérés par leur longitude (L) et leur latitude (l). Le satellite est repéré par sa position sur l'orbite géostationnaire et par longitude seule (Ls).

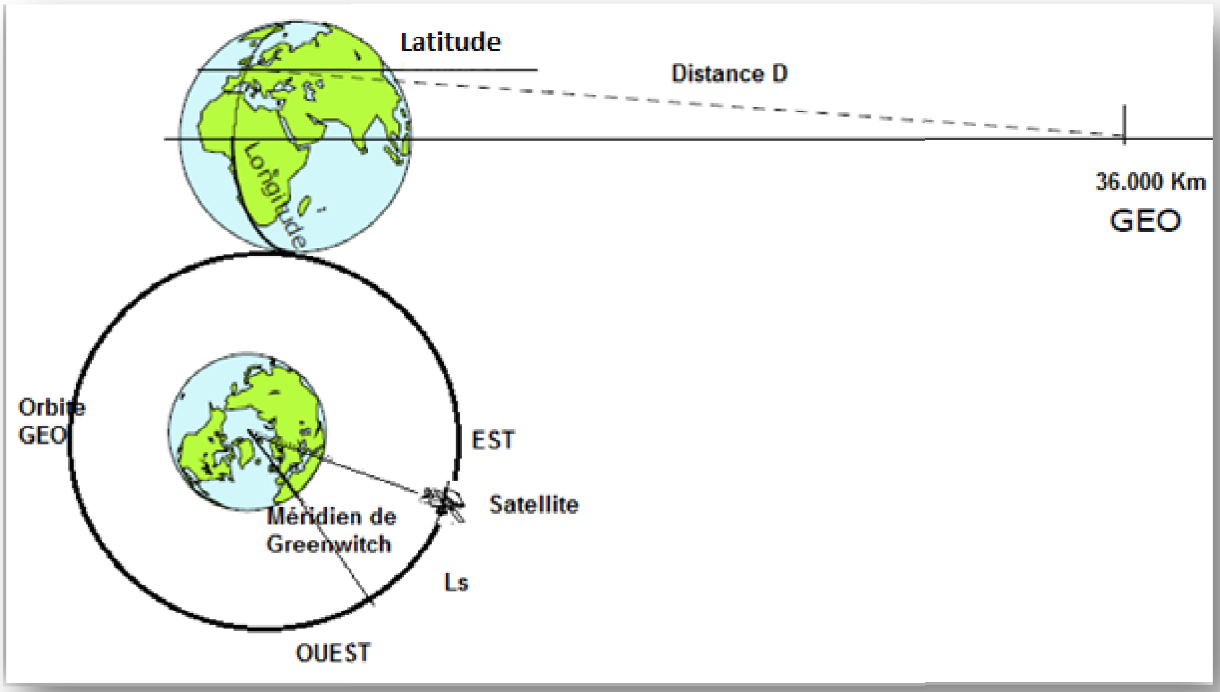


Figure III.4 : Distance station terrienne-satellite

Avec ces données on peut déterminer la distance D station terrienne-satellite présentée par la formule suivante :

$$D = r \sqrt{\quad + \ , \ (\ - \ (\ - \ ) \ (\ )} \qquad [Km] \ldots\ldots\ldots\ldots (5)$$

On à comme données :

r : le rayon de l'orbite géostationnaire au dessus du sol (r=35786Km).

L : longitude terrestre Est ou Ouest.

l : latitude terrestre Nord ou Sud.

Ls : longitude du satellite Est ou Ouest.

e. Azimut et élévation :

Le pointage de l'antenne consiste, d'une part à trouver le satellite, et d'autre part à obtenir un gain maximum.

L'antenne doit être pointée avec plus de précision possible vers le satellite dans le but d'obtenir le meilleur gain. En fait, ce pointage fait intervenir des paramètres tels que l'azimut et l'élévation qui sont en fonction de la latitude et la longitude.

La figure suivante nous présente les deux paramètres l'azimut et l'élévation.

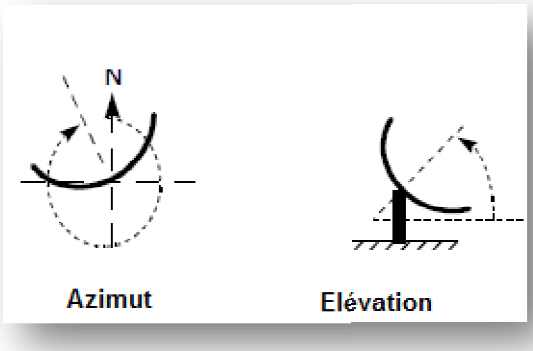


Figure III.5 : Azimut et Elévation

Les formules de l'Azimut et de l'élévation sont données comme suite :

$$\textbf{Elévation=Arc sin}\ \frac{+\quad -\ ^{2}}{} \qquad \text{(radian)}\ldots\ldots\ldots\ldots\ldots(6)$$

$$\textbf{Azimut=Arc sin}\ \frac{(\quad)}{(\quad)\ ^{2}(\ )} \qquad \text{(radian)}\ldots\ldots\ldots\ldots\ldots(7)$$

R : rayon terrestre au niveau de l'équateur (R=6378 Km).

Selon les besoins et les cas d'utilisation de ses paramètres, on va présenter sous forme d'un tableau les quelques conversion entre les unités utilisées :

Types d'Angle	La conversion
Angle en radian	$\frac{*\qquad\qquad é}{180}$
	$\frac{*}{200}$
Angle en grade	$\frac{200*\qquad\qquad é}{180}$

	$\frac{200 *}{}$
Angle en degré	$\frac{180 *}{}$
	$\frac{180 *}{200}$

Tableau III.1 : conversion Angle

III.4. Bilan de puissance :

III.4.1. Gain d'antenne :

Dans le calcul d'une liaison, le gain est la caractéristique la plus importante d'une antenne parabolique qui est généralement utilisée dans les systèmes de communication par satellite. Le gain peut être exprimé comme suite :

$$= (\quad (—) \quad) \text{……………………………………………………….……….(1)}$$

Avec, D est le diamètre de l'antenne, est la longueur d'onde et n est le rendement de l'ouverture de l'antenne (varie entre 50% et 70%).
Prise en dB, cette expression devient :

$$(G)_{dB}=10 \log (\quad (—) \quad) \qquad (dB)\text{……………………………..…….(2)}$$

III.4.2. EIRP (PIRE):

Le PIRE signifie la puissance Isotrope Rayonnée Equivalente et souvent notée EIRP (Equivalent sotropically Radiated Power). Elle représente la puissance équivalente émise par un amplificateur de puissance P0 associé à une antenne de Gain G. Elle se mesure en dBw.

$$EIRP_{dBw}=10 \log(G.P_0) \qquad (dBw)\text{……………. (3)}$$

Où, G est le gain d'antenne en dBi, et P0 est la puissance rayonnée en watts.

III.4.3. Facteur de mérite − :

On appelle le facteur de mérite le rapport du gain de l'antenne par la la température de bruit T d'un récepteur au niveau de la station de base

III.4.4. Rapport signal sur bruit au niveau satellite *(C/N)m :*

$$\frac{}{} \quad \text{m} = \frac{\quad}{\quad} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

PIREsol : PIRE de la station terrienne émettrice =puissance fournie à l'antenne Pe*Gain de l'antenne Ge.

(G/T)sat : facteur de mérite du satellite.

Lm : paramètre dépendant de la longueur d'onde λ du signal transmis et de la distance D station terrienne émettrice-satellite= $\frac{}{}^{2}$.

k : constante de Boltzmann=1.38 10 W/Hz/K.

B : bande passante équivalente de bruit.

Exprimé en dB nous avons : $\frac{}{}_{\text{m,dB}} = 10.\log_{10} \frac{}{}_{\text{m}}$

III.4.5. Rapport signal sur bruit au niveau station *(C/N)d:*

$$\frac{}{} \quad \text{d} = \frac{\quad}{\quad} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

PIREsat : PIRE du satellite =puissance fournie à l'antenne Psat*Gain de l'antenne Gsat.

(G/T)sol : facteur de mérite de la station sol.

Ld: paramètre dépendant de la longueur d'onde λ du signal transmis et de la distance D entre le satellite et la station terrienne réceptrice = $\frac{}{}^{2}$.

k : constante de Boltzmann=1.38 10 W/Hz/K.

B : bande passante équivalente de bruit.

Exprimé en dB nous avons : $\frac{}{}_{\text{d,dB}} = 10.\log_{10} \frac{}{}_{\text{d}}$

III.4.6. Rapport signal sur bruit global *:*

Le rapport signal sur bruit global est donné par la formule suivante :

$$\frac{}{} = \frac{1}{\frac{1}{\frac{}{}} + \frac{1}{\frac{}{}}}$$

Cette formule est en fait générale et s'applique pour toute liaison utilisant un transpondeur qu'il soit spatial ou terrestre.

En fait, est le rapport signal sur bruit en sortie du récepteur de la station terrienne. Il est utilisé pour la mesure de la qualité de la partie analogique d'un lien de communication. Cependant, dans le cas d'un lien de communication numérique, on introduit le BER (Bite Error Rate). Le BER est défini comme le nombre de bits erronés sur le nombre total de bits reçus. Le rapport BER est fonction de la qualité $_0$ avec Eb est l'énergie par bit er No est la densité de bruit du signal.

III.4.7. Quelques formules de Conversion:

a) Conversion vers :

$$= + (10 \log (\frac{}{})) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

D : est le débit.

OBW : bande passante occupée.

b) Conversion vers :

$$= -10 \log\left(\frac{}{}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

c) conversion dB :

On va présenter sous forme d'un tableau les quelques formules qui peuvent nous aider à faire les conversions dB nécessaires selon les besoins :

Type de Conversion	Formule de conversion
Conversion dBm vers dBW	dBW=dBm-30
Conversion dBW vers dBm	dBm=dBW+30
Conversion dBW vers les Watts	Watts=$10^{(\frac{}{})}$
Conversion Watts vers dBW	dBW=10*log(Watts)

Tableau III.2 : conversion dB.

Dans ce qui suit, nous montrons les interfaces d'un logiciel de calcul de tous les paramètres d'une liaison satellitaire sous Matlab.

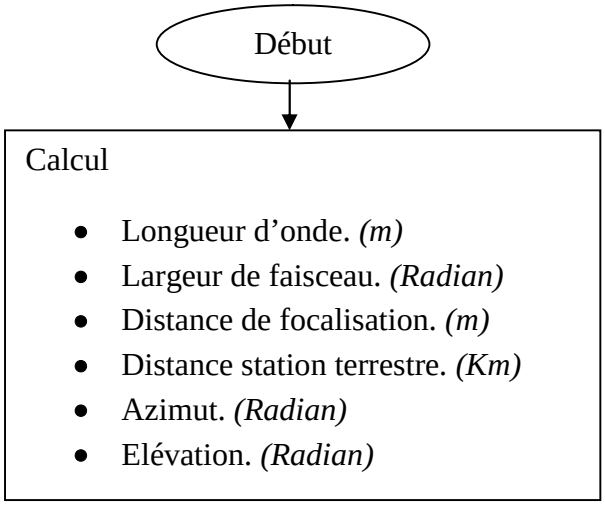
III.5.:Résultats et discussion :

Contrairement a la partie précédente, celle ci est réservée au calcul des paramètres de notre liaison sous Matlab, donc c'est une partie purement pratique.

Il est simple à utiliser car tous les paramètres du satellite sont déjà inclus. Il suffit de connaitre les caractéristiques de la station pour pouvoir l'utiliser. (Le programme se trouve dans l'annexe).

Nous utiliserons les diagrammes du calcul des bilans de notre liaison présentés dans les figures III.6, III.8, et III.11 pour bien comprendre la manipulation du logiciel que nous avons réalisé dont les résultats sont donnés sur des exemplaires comme les montres les figures III.7, III.9, III.12 en entrant les caractéristiques de la station, nous aurons automatiquement les résultats.

Paramètres d'antenne



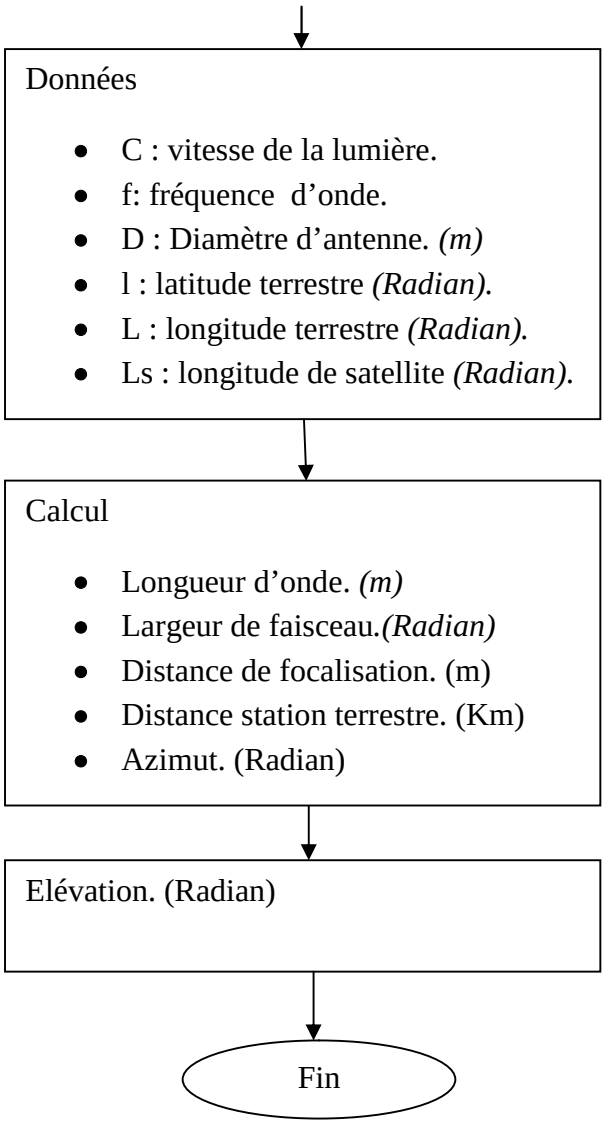


Figure III.6 : Diagramme de calcul des paramètres d'antenne.

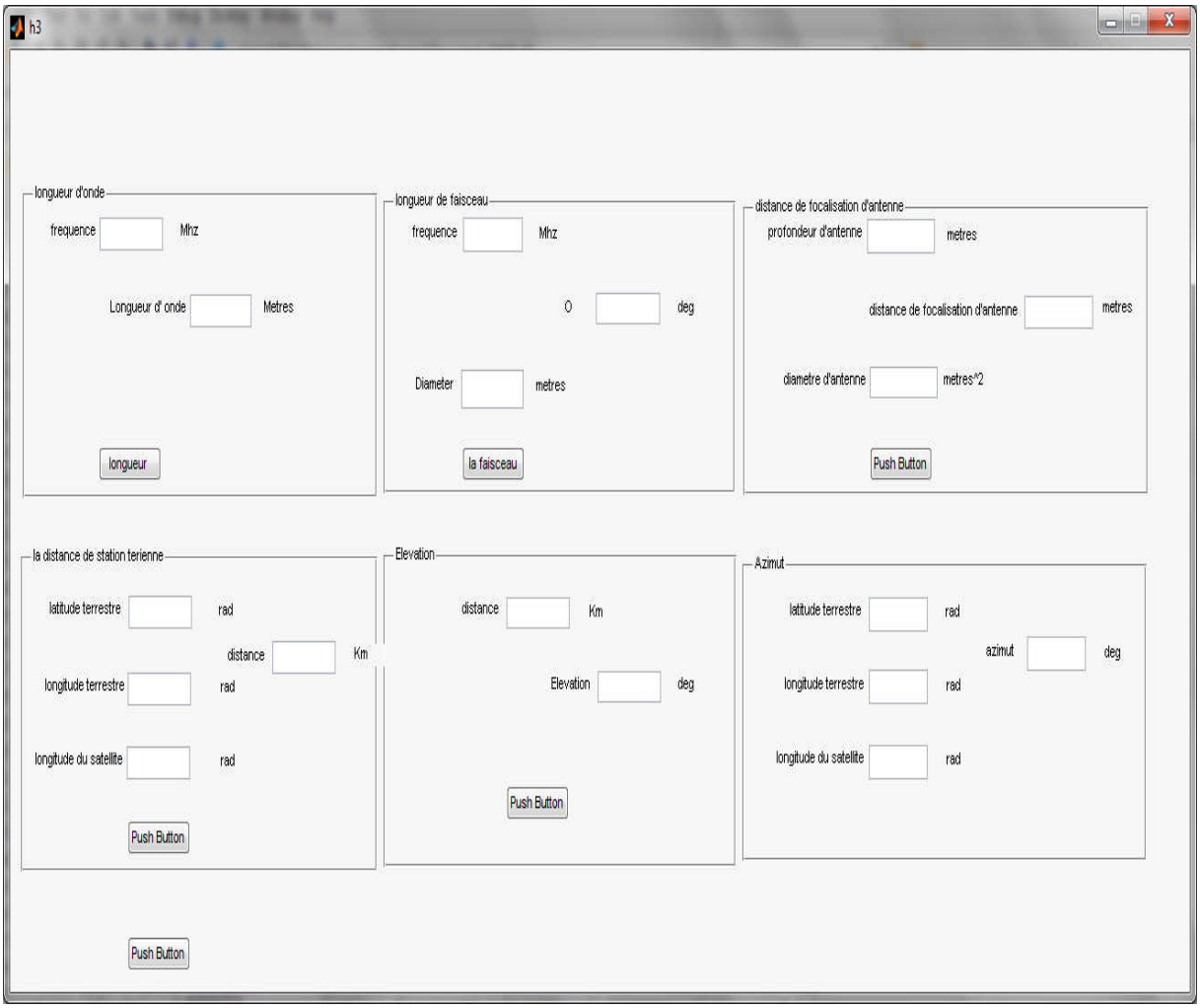


Figure III.7 : feuille de résultats de calcul des paramètres d'antenne.

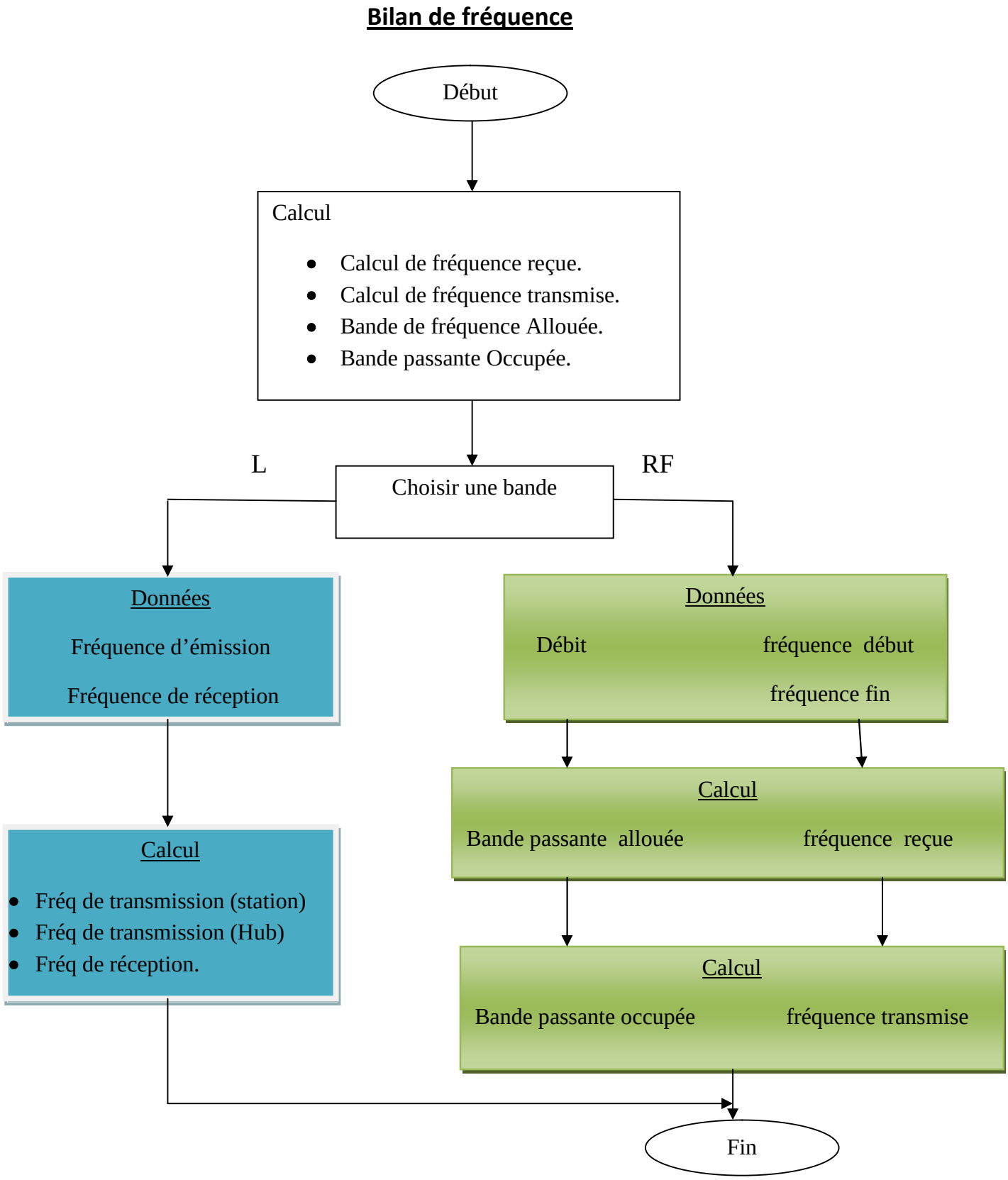


Figure III.8 : Diagramme du calcul de bilan de fréquence

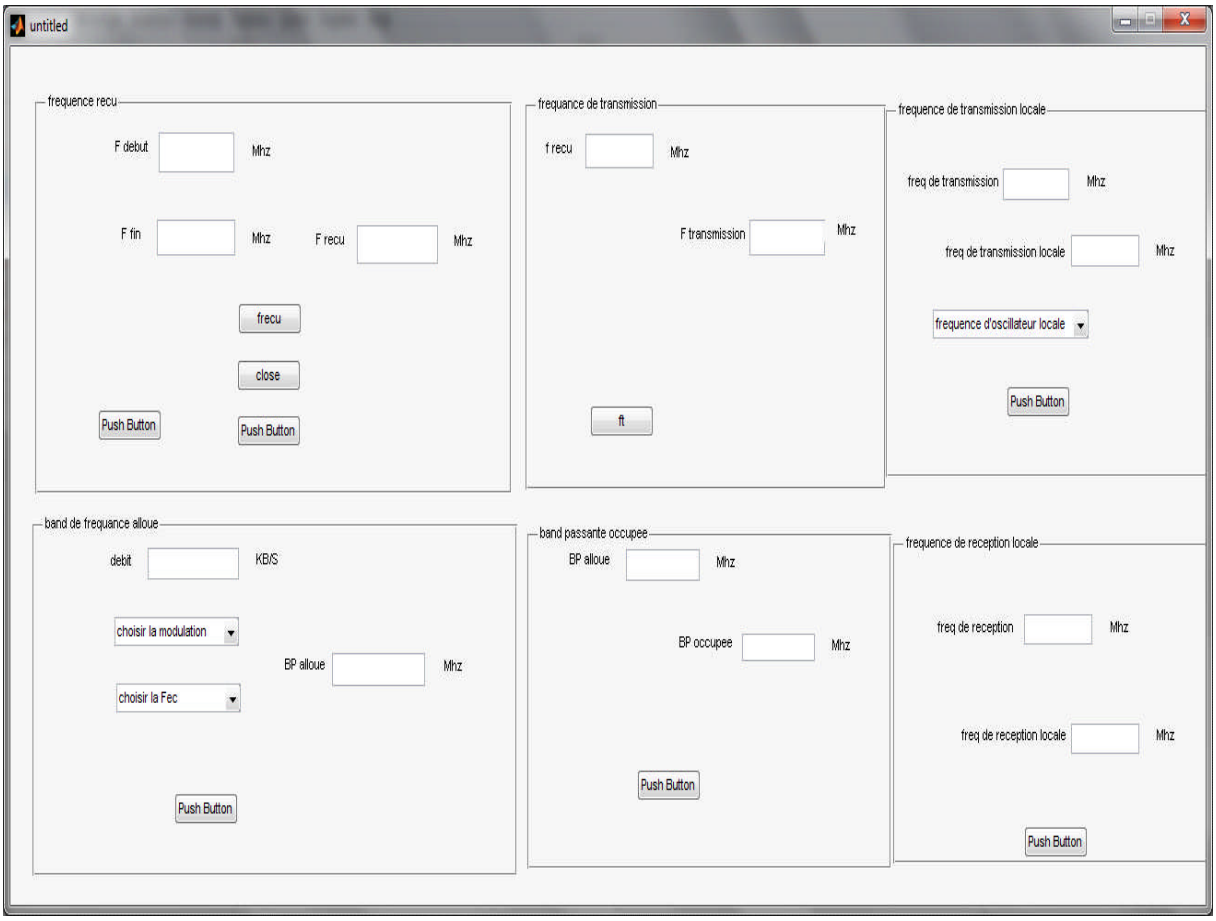


Figure III.9 : feuille de résultats du calcul de bilan de fréquence.

Pour vérifier les résultats de notre application, voici un bilan officiel des valeurs réelles de bilan de fréquence offert par Nedjma.

				Frequency				Modulation							
Transmit Site Code	ABIS (Transported sites)	Local Site Name	Ant. Size (m)	Pol Tx	Frequency Tx (MHz)	Pol Rx	Frequency Rx (MHz)	Data Rate (kbps)	Code Rate	Mod Dpsk, Bpsk, Other	Occ. BW (kHz)	Alloc BW (kHz)		Start Freq (MHz)	Stop Freq (MHz)
															11 459,0000
WAT01 E2	ALGER Hub	ALGER	4,9	V	14 260,1400	H	11 460,1400	4800	3/4	8PSK	2133	2 880,0	0,0000	11 459,0000	11 461,8800
IL3309	IL3309	BORDJ OMAR DRISS	2,4	V	14 261,9160	H	11 461,9160	120	3/4	8PSK	53	72,0	0,0000	11 461,8800	11 461,9520
TA1106	TA1106	IDLES	2,4	V	14 261,9880	H	11 461,9880	120	3/4	8PSK	53	72,0	0,0000	11 461,9520	11 462,0240
TA1104	TA1104	ADALISSA	2,4	V	14 262,0600	H	11 462,0600	120	3/4	8PSK	53	72,0	0,0000	11 462,0240	11 462,0960
TA1112	TA1112	TAZROUK	2,4	V	14 262,1320	H	11 462,1320	120	3/4	8PSK	53	72,0	0,0000	11 462,0960	11 462,1680
IL3308	IL3308	BORDJ EL HAOUES	2,4	V	14 262,2040	H	11 462,2040	120	3/4	8PSK	53	72,0	0,0000	11 462,1680	11 462,2400
TA1107	TA1107	TA1107	2,4	V	14 262,2760	H	11 462,2760	120	3/4	8PSK	53	72,0	0,0000	11 462,2400	11 462,3120
			2,4	V	14 262,3480	H	11 462,3480	120	3/4	8PSK	53	72,0	0,0000	11 462,3120	11 462,3840
IL3310	IL3310	DERDER	2,4	V	14 262,4200	H	11 462,4200	120	3/4	8PSK	53	72,0	0,0000	11 462,3840	11 462,4560
EXT01	Inbound Abis 120		2,4	V	14 262,4920	H	11 462,4920	120	3/4	8PSK	53	72,0	0,0000	11 462,4560	11 462,5280
OU3060	OU3060		2,4	V	14 262,5640	H	11 462,5640	120	3/4	8PSK	53	72,0	0,0000	11 462,5280	11 462,6000
GH4/16	GH4/16	EL GOLEA 1	2,4	V	14 262,6720	H	11 462,6720	240	3/4	8PSK	107	144,0	0,0000	11 462,6000	11 462,7440
OU3063	OU3063		2,4	V	14 262,8160	H	11 462,8160	240	3/4	8PSK	107	144,0	0,0000	11 462,7440	11 462,8880
IL3304	IL3303+IL3304	DJANET2 AEROPORT	2,4	V	14 262,9600	H	11 462,9600	240	3/4	8PSK	107	144,0	0,0000	11 462,8880	11 463,0320
IL3302	IL3302	DJANET1	2,4	V	14 263,1040	H	11 463,1040	240	3/4	8PSK	107	144,0	0,0000	11 463,0320	11 463,1760
DJ1736	DJ1736	DJELFA	2,4	V	14 263,2480	H	11 463,2480	240	3/4	8PSK	107	144,0	0,0000	11 463,1760	11 463,3200
OU3059	I		2,4	V	14 263,3920	H	11 463,3920	240	3/4	8PSK	107	144,0	0,0000	11 463,3200	11 463,4640
IL3300	ILLIZI	ILLIZI	2,4	V	14 263,5720	H	11 463,5720	360	3/4	8PSK	160	216,0	0,0000	11 463,4640	11 463,6800
TN3702	TN3702+TN3704	TINDOUF 1	2,4	V	14 263,7880	H	11 463,7880	360	3/4	8PSK	160	216,0	0,0000	11 463,6800	11 463,8960
TN3700	TN3700+TN3705+TN37	TINDOUF 2	2,4	V	14 264,0400	H	11 464,0400	480	3/4	8PSK	213	288,0	0,0000	11 463,8960	11 464,1840
IL3305	IL3305+IL3316	IN AMENAS1	2,4	V	14 264,3280	H	11 464,3280	480	3/4	8PSK	213	288,0	0,0000	11 464,1840	11 464,4720
IL3311	IL3311+IL3315	IN AMENAS 2	2,4	V	14 264,6520	H	11 464,6520	600	3/4	8PSK	267	360,0	0,0000	11 464,4720	11 464,8320
TA1101	TA1102+TA1101	TAMI	2,4	V	14 265,0840	H	11 465,0840	840	3/4	8PSK	373	504,0	0,0000	11 464,8320	11 465,3360

Figure III.10. Bilan officiel de Nedjma.

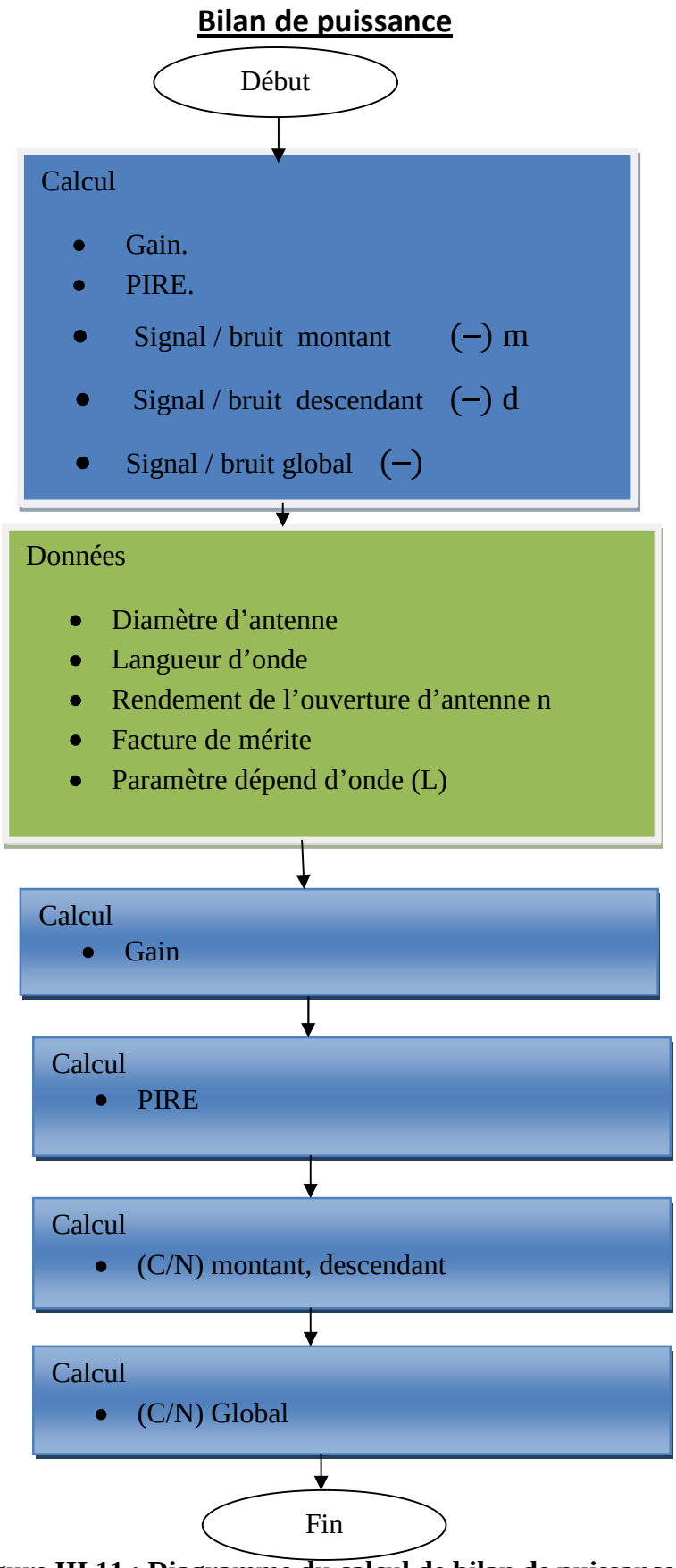


Figure III.11 : Diagramme du calcul de bilan de puissance.

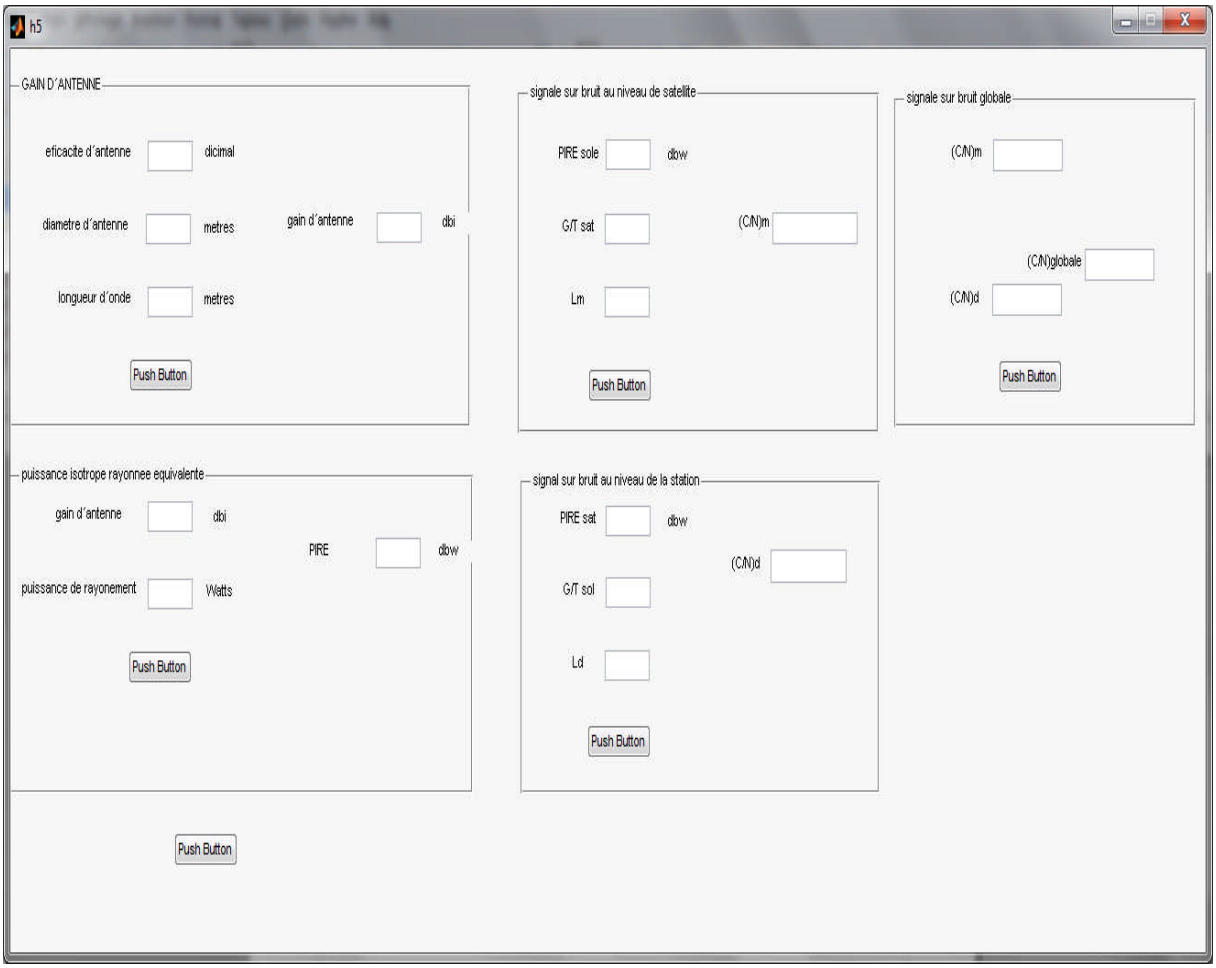


Figure III.12: feuille de résultats du calcul de bilan de puissance.

III.6. conclusion :

Ces quelques formules fondamentales permettent de mieux comprendre comment sont calculés les bilans entre deux stations terriennes et un satellite. Elles permettent de dimensionner les antennes et les émetteurs des satellites en fonction de la mission visée c'est-à-dire de la couverture, de l'accessibilité etc.….

CONCLUSION GENERALE

Les systèmes de télécommunications par satellites sont bien adaptés pour assurer, en complémentarité avec les réseaux terrestres, des services de télécommunications à la fois régionaux et mondiaux. Cette complémentarité doit se comprendre aussi bien en terme de sécurisation de réseaux terrestres qu'en terme de fourniture de services spécialisés dans des contextes économiquement moins avantageux par des moyens terrestres. Les satellites offrent l'avantage d'une couverture étendue favorisant les liaisons à longue distance, les liaisons entre sites multiples et la radio diffusion. L'immatérialité des ondes leur permet de s'affranchir des obstacles géographiques et offre un accès aisé aux zones déshéritées.

Ce mémoire créé pour devrait aider la compréhension du système Vsat par les techniciens et stagiaires voulant intégrer l'entreprise Nedjma ou même fournir aux entreprises clientes, non spécialisées dans les télécommunications, des explications globales sur la technologie qu'elles emploient ou qu'elles vont déployer.

Comme il l'a été démontré dans les chapitres précédents, la technologie Vsat permet de mettre en place différents réseaux : de données, téléphoniques, vidéos. Comme ces réseaux peuvent fonctionner en même temps, il faut bien prévoir tous les équipements nécessaires à chaque pour permettre l'utilisation des différents réseaux.

L'investissement étant relativement important au départ, l'étude préliminaire ne doit oublier aucun paramètre afin de rentabiliser au maximum le système une fois qu'il sera en production.

BIBLIOGRAPHIE

Sites internet :

1. http://familleOlivier.fr/regis/My%20eBooks/cours%20reseau/Cnam/cours%20reseaux/techniques/sat.htm.
2. http://djoumsergio.unblog.fr/tag/technologie-de-linformation-et-de-la-communication
3. http://Broadcast-Technology.com
4. http://www.amplus.com: site Web dédié aux équipements terrestres.
5. http://www.telesatellite. Com/articles/NetParSat/ : Site Web dédié à la réception satellite. Un dossier de quelques pages sur les terminaux Vsat.

Thèses :

6. Jean-Philippe MULLER, Le Réseau GSM et le Mobile.
7. Emmanuel Tonye & Landry Ewoussoua, Architecture GSM, GPRS et UMTS
8. M. Terré, Système De Communication, Satellite, v2.1
9. Livre Blanc, L'internet Par Satellite, (Avril 2010).
10. ALLOUCHE Benjamin & CHABAL Silvère, La communication de données par satellites
11. R.MEZARI & M.LAHDIR, Réseaux Locaux. Edition les pages bleus.
12. Gercel FLORES, Etude de La Technologie VSAT
13. Guide Vsat bidirectionnel, (septembre 2005) : un guide sur les réseaux Vsat.
14. Gérard Maral, Vsat networks (seconde édition),2003.Ecole Nationale Supérieure des télécommunications, Site de Toulouse France.
15. Olfa SAMET, Apport et complémentarité des satellites avec d'autres technologies terrestres émergentes, 2007.
16. Cours : Architecture protocolaire des réseaux mobiles BSS, Fabrice Valois, INSA de lyon. Avril 2004.

Annexe1: les programmes des paramètres sous Matlab

Paramètres d'antenne

```
function varargout = h3(varargin)
% H3 M-file for h3.fig
%      H3, by itself, creates a new H3 or raises the existing
%      singleton*.
%
%      H = H3 returns the handle to a new H3 or the handle to
%      the existing singleton*.
%
%      H3('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in H3.M with the given input arguments.
%
%      H3('Property','Value',...) creates a new H3 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before h3_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to h3_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only on
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help h3

% Last Modified by GUIDE v2.5 28-Jun-2011 19:26:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @h3_OpeningFcn, ...
                   'gui_OutputFcn',  @h3_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 h3 is made visible.
function h3_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 h3 (see VARARGIN)

% Choose default command line output for h3
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes h3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = h3_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 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


% --- 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)
```

```
freq=str2double(get(handles.edit1,'string'))
c=300
Londe=c/freq
set(handles.edit2,'string',Londe)



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 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)
FREQ=str2double(get(handles.edit3,'string'))
D=str2double(get(handles.edit4,'string'))
O=21/(FREQ*D)
set(handles.edit5,'string',O)
d=str2double(get(handles.edit6,'string'))
f=(D^2)/(16*d)
set(handles.edit7,'string',f)



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



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a
double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (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 edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a
double


% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (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)
D=str2double(get(handles.edit8,'string'))
d=str2double(get(handles.edit6,'string'))
f=(D^2)/(16*d)
set(handles.edit7,'string',f)



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



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


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
r=6378
R=35786
LS=str2double(get(handles.edit11,'string'))
L=str2double(get(handles.edit10,'string'))
l=str2double(get(handles.edit9,'string'))
d=r*sqrt(1+0.42*(1-cos(LS-L)*cos(l)))
set(handles.edit12,'string',d)



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



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (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 edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a
double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (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 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)

R=6378
r=35786
D=str2double(get(handles.edit13,'string'))
ELEV=asin(((r+2*R)*r-D^2)/(2*R*D))
set(handles.edit14,'string',ELEV)



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 calle
```

```
% 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



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a
double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (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 edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a
double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (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)
CLOSE
LIAISON


% --- 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)
L=str2double(get(handles.edit16,'string'))
l=str2double(get(handles.edit15,'string'))
LS=str2double(get(handles.edit17,'string'))
AZI=asin(sin(LS-L)/sqrt(1-cos(LS-L)^2*cos(l)^2))
set(handles.edit18,'string',AZI)
```

Bilan de fréquence

```
function varargout = h3(varargin)
% H3 M-file for h3.fig
%      H3, by itself, creates a new H3 or raises the existing
%      singleton*.
%
%      H = H3 returns the handle to a new H3 or the handle to
%      the existing singleton*.
%
%      H3('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in H3.M with the given input arguments.
%
%      H3('Property','Value',...) creates a new H3 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before h3_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to h3_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only on
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help h3

% Last Modified by GUIDE v2.5 28-Jun-2011 19:26:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @h3_OpeningFcn, ...
                   'gui_OutputFcn',  @h3_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 h3 is made visible.
function h3_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 h3 (see VARARGIN)

% Choose default command line output for h3
```

```
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes h3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = h3_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 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


% --- 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)

freq=str2double(get(handles.edit1,'string'))
c=300
Londe=c/freq
set(handles.edit2,'string',Londe)
```

```
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 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)
FREQ=str2double(get(handles.edit3,'string'))
D=str2double(get(handles.edit4,'string'))
O=21/(FREQ*D)
set(handles.edit5,'string',O)
d=str2double(get(handles.edit6,'string'))
f=(D^2)/(16*d)
set(handles.edit7,'string',f)



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



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a
double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (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 edit8 as text
```

```
%        str2double(get(hObject,'String')) returns contents of edit8 as a
double


% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (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)
D=str2double(get(handles.edit8,'string'))
d=str2double(get(handles.edit6,'string'))
f=(D^2)/(16*d)
set(handles.edit7,'string',f)



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



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


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
r=6378
R=35786
LS=str2double(get(handles.edit11,'string'))
L=str2double(get(handles.edit10,'string'))
l=str2double(get(handles.edit9,'string'))
d=r*sqrt(1+0.42*(1-cos(LS-L)*cos(l)))
set(handles.edit12,'string',d)



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



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (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 edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a
double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (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 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)

R=6378
r=35786
D=str2double(get(handles.edit13,'string'))
ELEV=asin(((r+2*R)*r-D^2)/(2*R*D))
set(handles.edit14,'string',ELEV)



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



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a
double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (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 edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a
double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (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)
CLOSE
LIAISON


% --- 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)
L=str2double(get(handles.edit16,'string'))
l=str2double(get(handles.edit15,'string'))
LS=str2double(get(handles.edit17,'string'))
AZI=asin(sin(LS-L)/sqrt(1-cos(LS-L)^2*cos(l)^2))
set(handles.edit18,'string',AZI)
```

Bilan de puissance

```
function varargout = h5(varargin)
% H5 M-file for h5.fig
%      H5, by itself, creates a new H5 or raises the existing
%      singleton*.
%
%      H = H5 returns the handle to a new H5 or the handle to
%      the existing singleton*.
%
%      H5('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in H5.M with the given input arguments.
%
%      H5('Property','Value',...) creates a new H5 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before h5_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to h5_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 h5

% Last Modified by GUIDE v2.5 27-Jun-2011 11:52:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @h5_OpeningFcn, ...
                   'gui_OutputFcn',  @h5_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 h5 is made visible.
function h5_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 h5 (see VARARGIN)

% Choose default command line output for h5
```

```
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes h5 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = h5_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 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)

eff=str2double(get(handles.edit1,'string'))
D=str2double(get(handles.edit2,'string'))
l=str2double(get(handles.edit3,'string'))
G=10*log10(eff*((pi*D)/l)^2)
set(handles.edit4,'string',G)



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



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a
double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (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 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)
G=str2double(get(handles.edit5,'string'))
P=str2double(get(handles.edit6,'string'))
PIRE=10*log10(G*P)
set(handles.edit7,'string',PIRE)



function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (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 edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (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 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



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


% --- 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
K=1.38*10^(-17)
B=3
PIRE=str2double(get(handles.edit8,'string'))
G=str2double(get(handles.edit9,'string'))
Lm=str2double(get(handles.edit10,'string'))
C=(PIRE*G*Lm)/(K*B)
set(handles.edit11,'string',C)



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
```

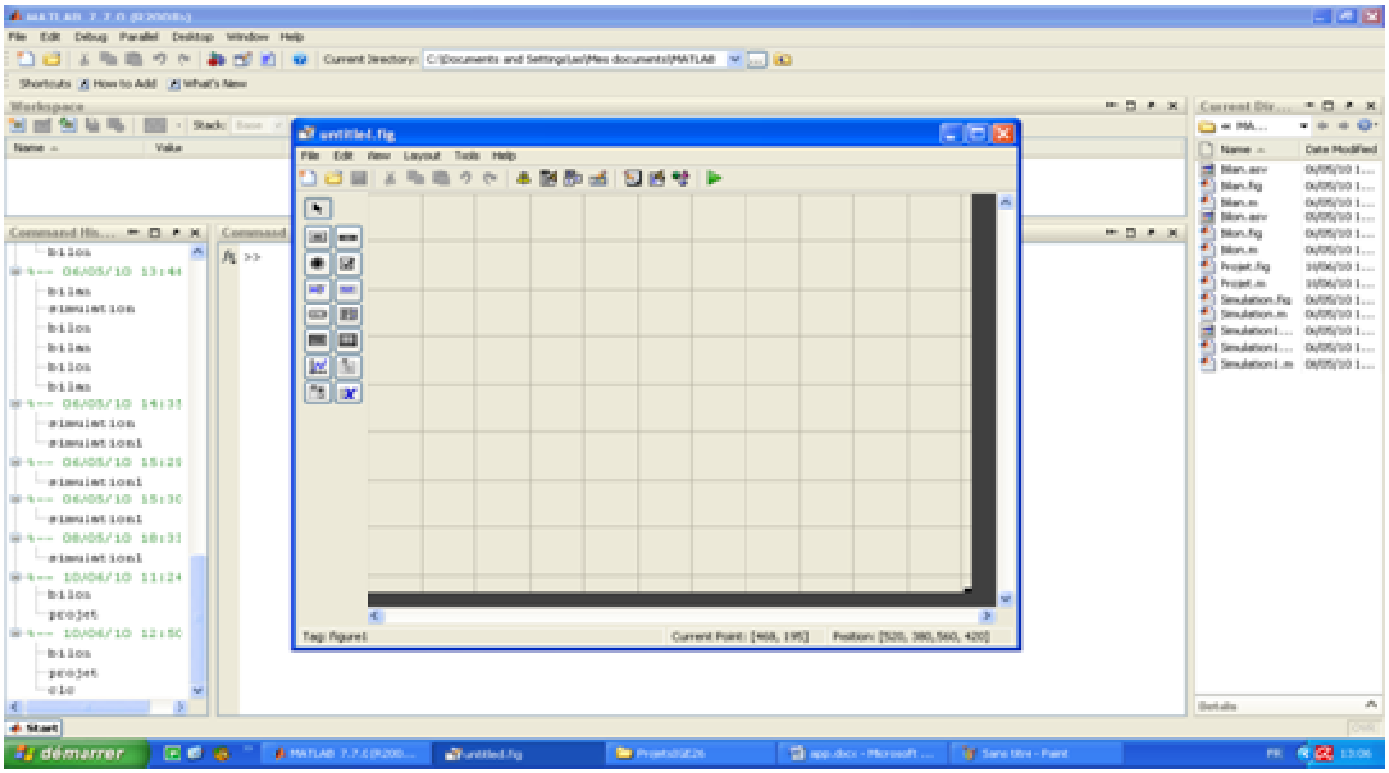
```
% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns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



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (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 edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a
double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

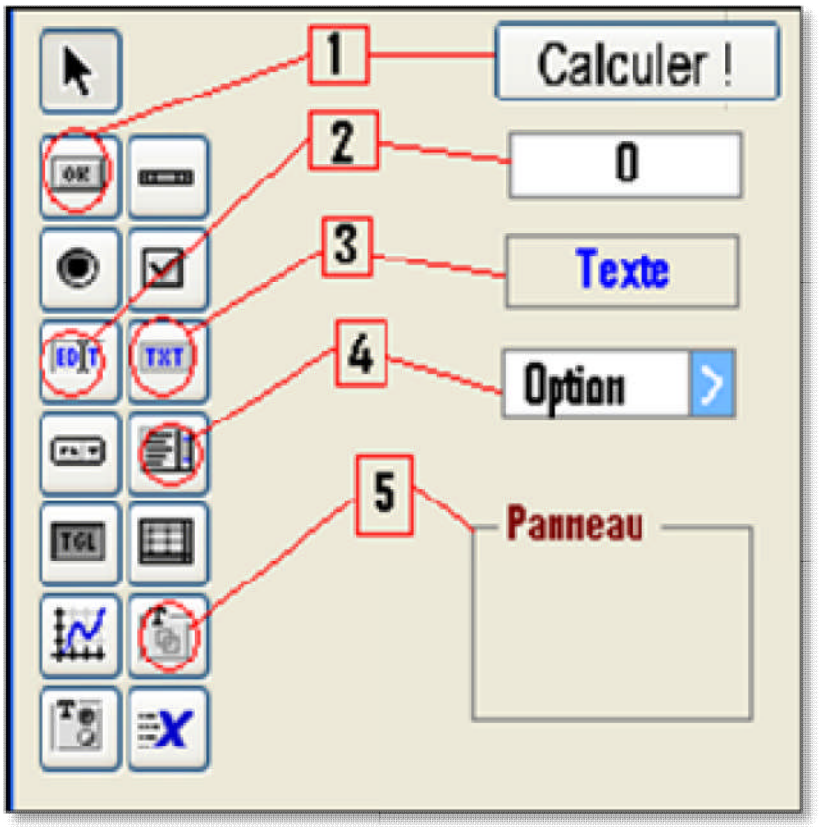
```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function 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


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
K=1.38*10^(-17)
B=3
PIRE=str2double(get(handles.edit12,'string'))
G=str2double(get(handles.edit13,'string'))
Lm=str2double(get(handles.edit14,'string'))
C=(PIRE*G*Lm)/(K*B)
set(handles.edit15,'string',C)



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
```

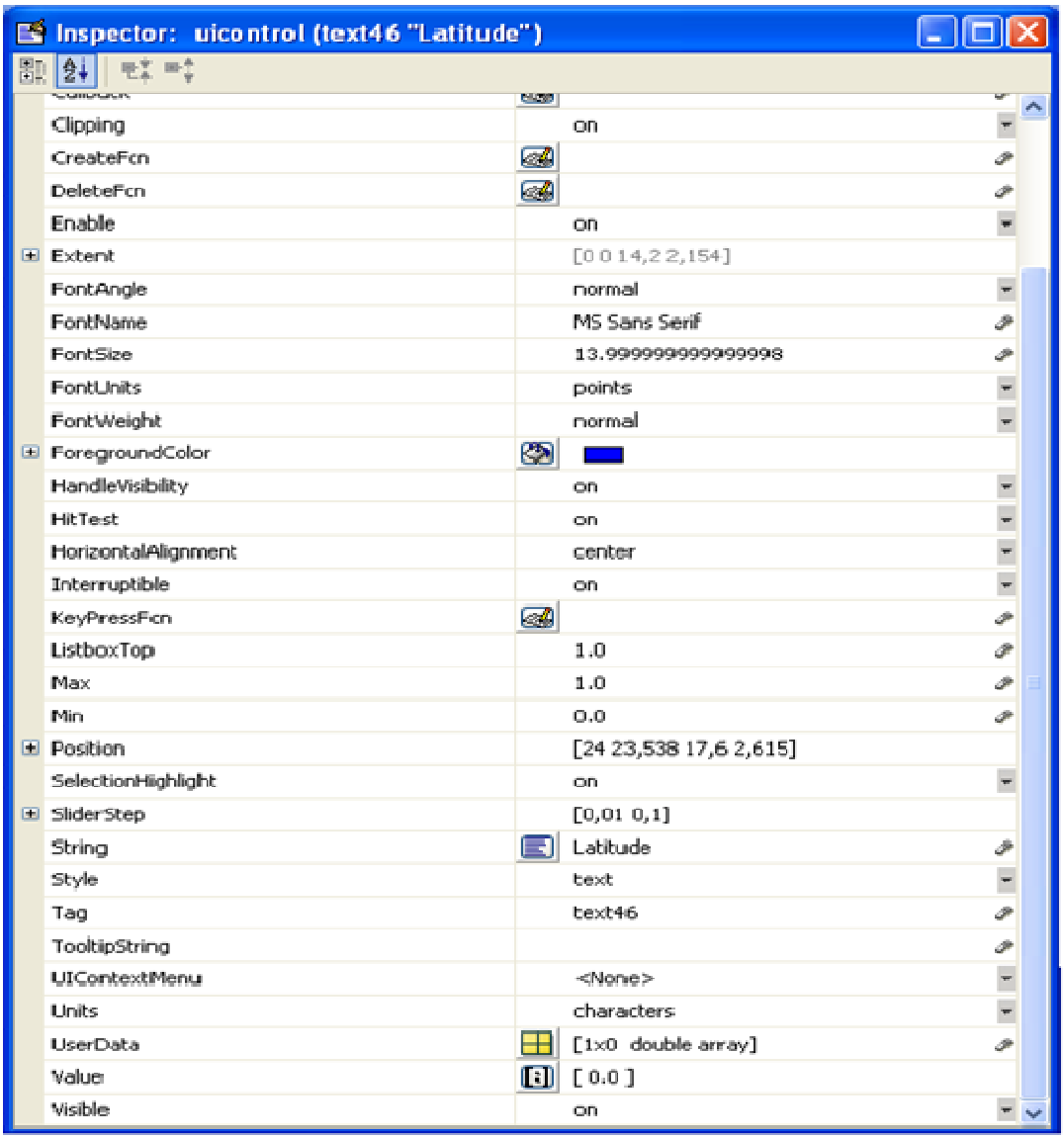
```
% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns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



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a
double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (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 edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a
double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (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 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)
cm=str2double(get(handles.edit16,'string'))
cd=str2double(get(handles.edit17,'string'))
cglobale=1/((1/cm)+(1/cd))
set(handles.edit18,'string',cglobale)


% --- 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)
close
LIAISON
```

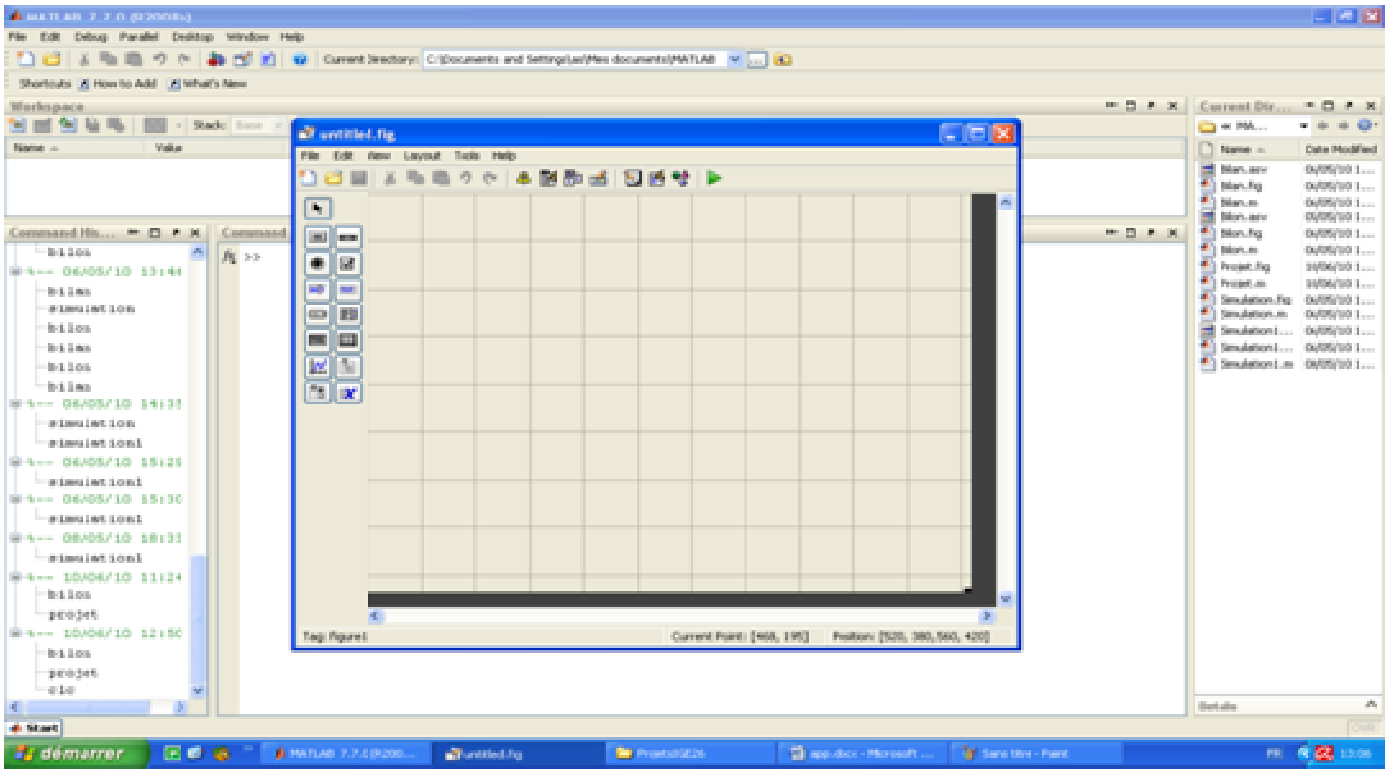
Annexe2- les 1ères étapes à suivre pour créer l'interface :

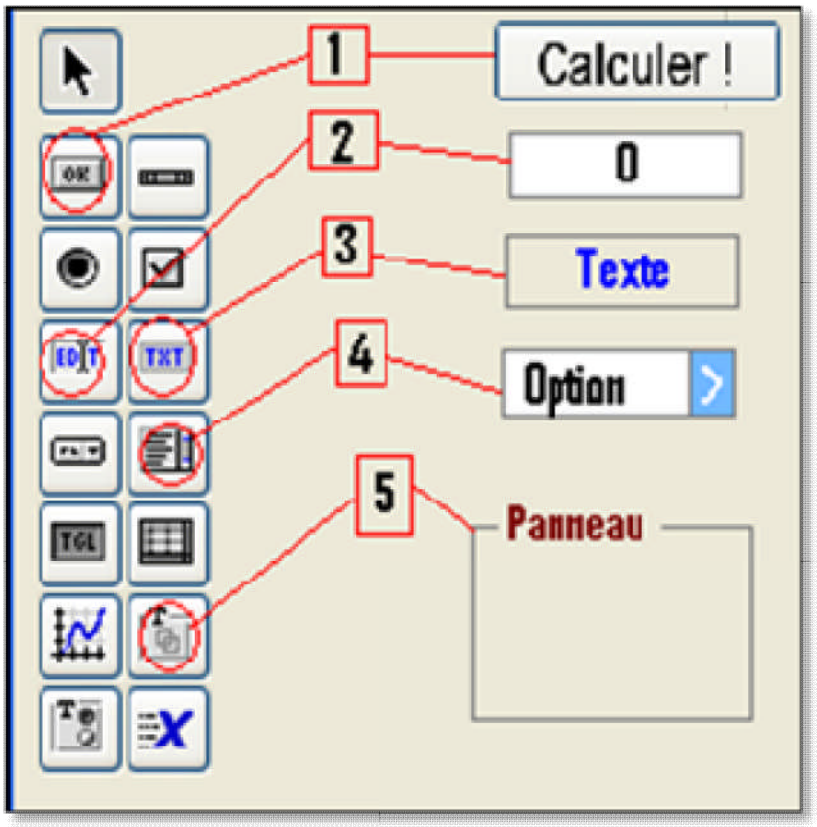


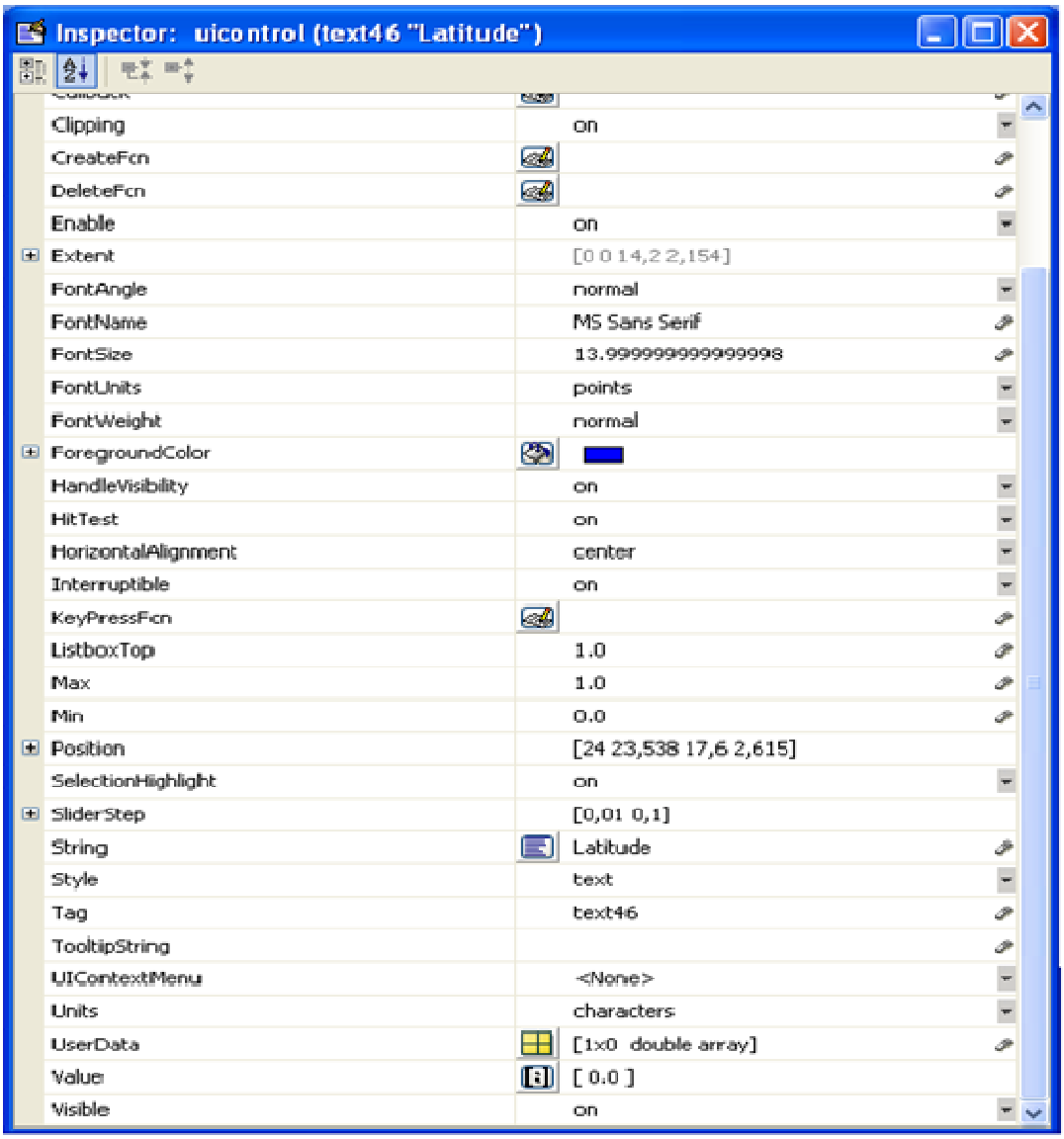



Inspector: uicontrol (text46 "Latitude")
Clipping on
CreateFcn
DeleteFcn
Enable on
Extent [0 0 14,2 2,154]
FontAngle normal
FontName MS Sans Serif
FontSize 13.999999999999998
FontUnits points
FontWeight normal
ForegroundColor
HandleVisibility on
HitTest on
HorizontalAlignment center
Interruptible on
KeyPressFcn
ListboxTop 1.0
Max 1.0
Min 0.0
Position [24 23,538 17,6 2,615]
SelectionHighlight on
SliderStep [0,01 0,1]
String Latitude
Style text
Tag text46
TooltipString
UIContextMenu <None>
Units characters
UserData [1x0 double array]
Value [0.0]
Visible on

Annexe2- les 1ères étapes à suivre pour créer l'interface :






Inspector: uicontrol (text46 "Latitude")
Clipping on
CreateFcn
DeleteFcn
Enable on
Extent [0 0 14,2 2,154]
FontAngle normal
FontName MS Sans Serif
FontSize 13.999999999999998
FontUnits points
FontWeight normal
ForegroundColor
HandleVisibility on
HitTest on
HorizontalAlignment center
Interruptible on
KeyPressFcn
ListboxTop 1.0
Max 1.0
Min 0.0
Position [24 23,538 17,6 2,615]
SelectionHighlight on
SliderStep [0,01 0,1]
String Latitude
Style text
Tag text46
TooltipString
UIContextMenu <None>
Units characters
UserData [1x0 double array]
Value [0.0]
Visible on

Résumé

De nos jours, nous remarquons que les satellites prennent de plus en plus de place dans les télécommunications et leur fonction a changé avec les décennies. Parallèlement au déclin relatif de leur rôle dans la téléphonie, ils ont pris une importante croissance dans la télédiffusion et dans la diffusion de données. Ils amorcent un retour médiatisé dans la téléphonie.

Les satellites sont des éléments indispensables d'une toile spatiale qui recouvre de plus en plus le globe terrestre et qui aujourd'hui, permet des communications de n'importe quel point de la planète.

La technologie de transmission bidirectionnelle existe depuis les années 1980, elle est connue sous le nom de Vsat (Verry Small Aperture Terminal). Au tout début, ces réseaux Vsat étaient réservés à des environnements professionnels spécifiques tels que les plates-formes pétrolières, les camps d'expédition, les bases polaires .etc. puis les Vsat se sont généralisés au niveau des entreprises qui souhaitaient déployer des réseaux dans les zones délaissées ou inaccessibles par les opérateurs. Les tarifs des équipements et des services Vsat pour les entreprises restent relativement élevés et ne sont donc pas accessibles aux particuliers.

L'installation d'une station Vsat nécessite une étude préalable afin de rentabiliser au maximum le système une fois installé. L'entreprise dispose des différents guides d'installation spécifiques à chaque équipement. Cependant la création d'une procédure générale d'installation d'une station terrestre, après l'étude de la technologie Vsat faite dans ce rapport, peut aider un technicien désirant intégrer la société ou un stagiaire à mieux comprendre et d'acquérir les bases nécessaires pour bien accomplir sa tâche.

Dans cette étude, nous allons présenter les concepts et les éléments clés pour la compréhension des technologies satellites et plus particulièrement la technologie Vsat afin de déterminer les spécificités de ce type de technologie de transmission.

Ce mémoire créé pour devrait aider la compréhension du système Vsat par les techniciens et stagiaires voulant intégrer l'entreprise Nedjma ou même fournir aux entreprises clientes, non spécialisées dans les télécommunications, des explications globales sur la technologie qu'elles emploient ou qu'elles vont déployer.