Imagerie ultrasonore

Année 2022-2023
barbara.nicolas@creatis.insa-lyon.fr
herve.liebgott@creatis.insa-lyon.fr

TP : simulations des échographies ultrasonores

L'objectif final de ce TP est la construction d'une image échographique ultrasonore. Deux étapes seront nécessaires :

i. Conception de l'expérience et acquisition des signaux
ii. Traitement des signaux et formation de l'image

Le logiciel de simulation *Field_II* (http://field-ii.dk), développé par J.A. Jensen de l'université technique du Danemark (http://www.dtu.dk/centre/cfu/English.aspx), sera utilisé pour simuler la propagation des ondes ultrasonores dans le milieu étudié et ainsi générer les signaux radiofréquences qui seront ensuite traités par vos soins.

Le guide utilisateur de ce logiciel répertoriant l'ensemble des fonctions matlab utilisables est disponible en ligne. Quelques-unes des fonctions qui vous seront utiles sont également décrites en annexe de ce document.

Les milieux à imager peuvent être assimilés à des images codées en niveau de gris, appelées *fantôme numériques*, représentant une distribution spatiale de l'impédance acoustique du milieu. L'objectif de l'activité est la construction d'images échographiques aussi fidèle que possible à ces fantômes.

I – Etude de la source

a) Construire une antenne d'émission linéaire

On va ici définir l'antenne source à partir de laquelle les ondes acoustiques seront émises dans le milieu. Cette antenne prend la forme d'une mise en série de petits éléments piézo-électriques qui assurent la fonction de transducteur. Lorsque chaque élément est positionné de façon régulière le long d'une droite, l'antenne est dite linéaire.

Les points suivants sont définis :

- la géométrie de l'antenne (fonction `xdc_linear_array`) incluant le nombre d'éléments, leur taille et leur position respective,
- le signal d'excitation (fonction `xdc_excitation`) qui sera donné en entrée de chaque élément. Ce signal, identique pour tous les éléments dans un premier temps, comportera une fréquence centrale `nu0` (du domaine des ultrasons, rappel de l'ordre du

MHz). La fréquence d'échantillonnage `nus` des signaux est choisie en bon accord (prenez la fréquence avec Shannon et multipliez là au moins par deux).

- la réponse impulsionnelle des éléments (fonction `xdc_impulse`). Précisez le signal émis en sortie d'un transducteur. Cette réponse permet de fixer, entre autres, la bande passante du transducteur.

Etudier la fonction matlab `DefineTransducer.m` : les différents paramètres définis ainsi que les actions réalisées pour préparer l'émission des signaux.

b) Observer le champ de pression généré

i- Visualisation : La fonction `calc_hp` permet de mesurer en un ou plusieurs points le signal ultrasonore généré par la source.
Apprendre à :

- Observer le signal mesuré en un unique point de l'espace
- Observer les signaux mesurés le long de droites parallèles aux axes x et z

Ouvrir et étudier la fonction matlab `PressureField.m`

ii- Etapes de la création d'un faisceau ultrasonore :

La grandeur `focus` permet de focaliser « physiquement » la sonde. Il s'agit du paramètre de focalisation mécanique, i.e. les lentilles acoustiques présentes en face avant de la sonde. Lors de la création d'une image, ce paramètre ne change pas. Afin de ne pas y être sensible, on focalise simplement à l'infini les éléments, par exemple à 30 mètres, ce qui évite toute focalisation mécanique.

La focalisation électronique va être réalisée numériquement à l'aide de retards, on abodre cette question dans le paragraphe suivant.

Enfin, afin de limiter les effets de bords observés, une fenêtre d'apodisation peut être appliquée à l'antenne. Cette fenêtre consiste à limiter la contribution des éléments situés aux extrémités de l'antenne. On utilisera la fonction `xdc_apodization` avec la syntaxe :

```
>> apo = hanning(Nelements)';
>> xdc_apodization (emit_aperture, 0, apo);
```

c) Focalisation en émission

Commencez par régler le point de focus mécanique de votre transducteur à l'infini (typiquement `[x0,y0,z0] = [0,0,30]` mètres).

1- Focalisation par ligne à retards sur les transducteurs :
Réalisez une focalisation des ondes émises par la source sur le point focal de votre choix en définissant des retards à appliquer sur chaque élément du transducteur. Ainsi les transducteurs auront des instants d'émission différents, permettant la focalisation sur le point focal choisi.

Afin de calculer les retards à appliquer, la fonction xdc_get vous permet de récupérer les coordonnées de chaque élément de votre antenne. Vous pouvez utiliser le code suivant :

```
>> infos = xdc_get(emit_aperture,'rect');
>> center_points = infos(8:10,:)';
```

Note : Afin de focaliser en un point on doit compenser les différences de temps de vol entre chaque élément et le point focal. Ceci permet aux contributions de chaque élément d'interférer de manière constructive au point focal. Chaque outil de simulation, chaque système peut avoir une convention différente pour définir les retards (positifs, négatifs, référence au milieu etc…). Dans Field II, pour le calcul des retards, faire en sorte que le retard maximal soit nul et que les retards soient tous négatifs. Cela revient à ne pas décaler le signal envoyé par l'émetteur en face (au sens dimension latérale `x`) du point de focalisation. N'hésitez pas à tracer les retards calculés pour vous assurer de la forme de la fonction.

Une fois les retards calculés, utilisez la fonction `xdc_times_focus` pour les appliquer sur chaque élément de l'antenne. En notant `delays` le vecteur ligne des retards, on utilise la syntaxe suivante :

```
>> xdc_times_focus(emit_aperture,0,delays);
```

Vérifier que vous avez focalisé au point choisi à l'aide de la fonction `calc_hp`
Tester sur le point `focus_point = [10 0 30]/1000;`

2- Vérification de la focalisation

Cette focalisation en émission peut également être obtenue directement avec le logiciel Field à partir de la fonction `xdc_focus`

Exemple :

```
focus_point = [10 0 30]/1000;
xdc_center_focus (emit_aperture,[focus_point(1) 0 0]); % permet de
spécifier le point central par rapport auquel est fait la focalisation =
point de référence qui ne sera pas retardé
xdc_focus (emit_aperture, 0, focus_point);
```

Vérifier que les deux focalisations sont équivalentes.
Par la suite, utiliser l'une ou l'autre des méthodes pour focaliser à l'émission.

d) Optimiser les paramètres de l'antenne émettrice (à faire rapidement)

Faire évoluer les paramètres de votre source (nombre d'éléments, fréquence des signaux) afin de modifier la focalisation des ondes ultrasonores sur le point de focus de l'antenne. Commenter l'influence des deux paramètres.

II – analyse en réception

On s'intéresse dans cette seconde partie aux signaux mesurés au niveau des transducteurs après propagation dans le milieu et réflexion sur un ou plusieurs diffuseurs (« scatterers »).

a) Mise en place de l'antenne réceptrice

Bien qu'en pratique la même antenne de transducteurs assure les fonctions d'émission et de réception, le logiciel de simulation *Field_II* nécessite la création d'une seconde antenne qui servira de récepteur. Pour créer cette seconde antenne, utilisez la même syntaxe et les mêmes paramètres que pour l'antenne d'émission, à savoir :

```
>> receive_aperture = xdc_linear_array (Nelements, width, height, kerf, 1, 1, focus);
```

Ne pas oublier d'affecter la réponse impulsionnelle de l'antenne de réception avec la commande :

```
>> xdc_impulse(receive_aperture, rep_impuls_antenne );
```

avec `rep_impuls_antenne` la réponse impulsionnelle définie pour la sonde (même RI en émission et en réception)

Cette nouvelle antenne sera géométriquement confondue avec l'antenne émettrice, mais possèdera ses propres mémoires pour stocker les signaux mesurés.
Dans un premier temps, l'antenne d'émission reste identique à celle utilisée dans la partie I.b, avec le point de focus placé sur l'infini.

Définir maintenant un premier diffuseur en spécifiant ses coordonnées (x,y,z) en mètre ainsi que son amplitude. On écrira par exemple :

```
>> scatter_position = [0 0 50]/1000;
>> scatter_amplitude = [1];
```

Ouvrir le fichier `analyse_reception.m` qui contient quelques commandes pour faciliter votre progression dans le tp.

La fonction `calc_scat_multi` décrite en annexe vous permet de simuler la propagation dans le milieu et d'évaluer les signaux reçus au niveau de chaque transducteur de l'antenne de réception. Observer les signaux reçus pour différentes positions du diffuseur.

Ajouter un second diffuseur dans le milieu. Observer les signaux reçus pour différentes positions de ces diffuseurs et pour différentes amplitudes du diffuseur.

b) Focalisation à la réception

La focalisation à la réception consiste à appliquer un jeu de retards sur les signaux reçus au niveau de l'antenne de réception de façon à épouser la forme du front d'onde provenant d'un diffuseur choisi. A chaque diffuseur dans le milieu 2D correspond un unique jeu de retards au niveau de l'antenne. Une fois ce jeu de retards appliqué sur les signaux, leur sommation devient constructive en cas de présence d'un point diffractant, ce qui améliore considérablement la détection de ce point vis-à-vis du bruit d'une part et des autres diffuseurs d'autre part.

Focalisation par ligne à retards sur le transducteur :
En vous inspirant des fonctions développées dans la partie I.c.2, réalisez une focalisation à la réception sur le diffuseur de votre choix.

Pour cela, utiliser la fonction `xdc_center_focus` et `xdc_focus`, on utilise la syntaxe suivante :

```
>> focus_point = [10 0 30]/1000;
>> xdc_center_focus (receive_aperture,[focus_point(1) 0 0]);
>> xdc_focus (receive_aperture, 0, focus_point);
```

Réaliser ensuite la sommation des signaux retardés qui représente le résultat de votre formation de voies pour la ligne `x=x0`.
Observer le résultat.

Vérifier ce qui se passe si on focalise sur un point où il n'y a pas de diffuseur.

c) Focalisation en émission et en réception

Définir cinq diffuseurs de même amplitude et régulièrement répartis le long de l'axe z. On utilisera par exemple la syntaxe :

```
>> Npoints = 5;        %  Number of scatters
>> dz = 10/1000;       %  Distance between scatters [m]
>> zstart = 30/1000;   %  Starting point [m]
>> position = [zeros(1,Npoints);zeros(1,Npoints);(0:Npoints-1)*dz+zstart]';
>> amplitude = ones(Npoints,1);
```

Mettre en place la focalisation à l'émission et à la réception.
Permettre d'avoir le choix du point focal en émission et en réception séparément.

Comparer l'amplitude des signaux reconstruits après focalisation en réception sur le diffuseur central avec ou sans focalisation à l'émission sur ce même point.

III – Formation des images échographiques

a) Balayage linéaire

La formation d'une image échographique se réalise ligne par ligne au moyen d'un balayage du faisceau ultrasonore. Celui-ci peut-être linéaire (balayage suivant x) ou sectoriel (balayage suivant un angle d'émission).

Ouvrir étudier et adapter le programme Balayage_final.m qui permet de réaliser un balayage linéaire.

Transformer l'échelle de temps en échelle des profondeurs et vérifier les profondeurs des diffuseurs.

Vérifier la bonne localisation des diffuseurs sur l'image.

Pour visualiser une image « comme sur un échographe » appliquer la fonction rf2bmode a votre image.

- *Facultatif :* Un balayage sectoriel conduit à une acquisition en éventail. Adapter votre code ainsi que la visualisation à ce type de géométrie est un excellent exercice pour prolonger/approfondir les connaissances.

b) Traitements particuliers

i- Détection d'enveloppe : les signaux émis dans le milieu sont généralement centrés autour d'une fréquence porteuse (ou fréquence centrale) qui gêne la visualisation de l'image formée par la concaténation des signaux reçus. On cherche donc à éliminer les oscillations correspondantes à cette fréquence porteuse en ne retenant que l'enveloppe des signaux.

Observer l'enveloppe obtenue d'après le module du signal analytique, soit selon la syntaxe :

```
>> env = abs(hilbert(x));
```

ii- Compression logarithmique : face à la forte dynamique (écart entre le min et le max) des signaux radiofréquences, on code généralement les valeurs suivant une échelle logarithmique. Ceci permet notamment de s'adapter à la dynamique des écrans de visualisation.

Facultatif : Réalisez cette compression logarithmique pour une dynamique de 40 dB. Vous coderez ensuite les valeurs entre 0 et 127 (codage sur 7 bits) et afficherez les images en niveau de gris sur 128 valeurs (fonction `colormap(gray(128))`).
La fonction **rf2log** fournie réalise cette transformation. Utilisez la pour visualiser le milieu avec 5 diffuseurs.

iv- *Facultatif :* ***Interpolation :*** le nombre de lignes réalisées lors du balayage n'étant souvent pas suffisant pour une visualisation confortable des images échographiques, on a recours à une interpolation de l'image suivant l'axe x. La fonction `interp` de matlab permet de faire cette interpolation :

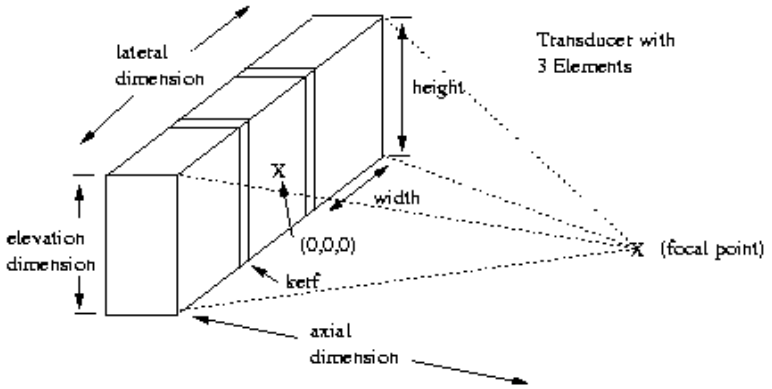
```
>> ID = 20; % facteur d'interpolation
>> for i=1:n
>>   img_interp(i,:) = interp(img(i,:),ID);
>> end
```

Annexe : Eléments d'utilisation du logiciel *Field II* avec matlab

Initialisation sous matlab (à ne faire qu'une fois)
>> field_init ;

Construire un transducteur
Définir la géométrie du transducteur (voir figure) : fonction `xdc_linear_array`

```
>> emit_aperture = xdc_linear_array (Nelements, width, height, kerf, 1, 1,
focus);
emit_aperture : pointeur vers le transducteur créé
Nelements : nombre d'éléments piézo-électriques de l'antenne
width : largeur des éléments suivant x
height : hauteur des éléments suivant y
kerf : distance entre les éléments suivant x
1, 1, : nombre de subdivisions dans chaque éléments pour une meilleur
        précision dans la résolution numérique des équations de
        propagation – Ne pas modifier.
focus : coordonnées (x,y,z) du point de focus initial du transducteur
```



Définir la réponse impulsionnelle : fonction `xdc_impulse`

```
>> xdc_impulse (emit_aperture, g);
emit_aperture : pointeur vers le transducteur concerné
g : réponse impulsionnelle sous forme d'un vecteur d'échantillons
```

Définir le signal d'excitation : fonction `xdc_excitation`

```
>> xdc_excitation (emit_aperture, e);
emit_aperture : pointeur vers le transducteur concerné
e : signal d'excitation sous forme d'un vecteur d'échantillons
```

Spécifier la fréquence d'échantillonnage : fonction `set_sampling`

```
>> set_sampling(nus);
nus : fréquence d'échantillonnage en Herz
```

Calculer le champ de pression généré par une antenne

On utilisera la fonction `calc_hp`

```
>> [hp,start_time] = calc_hp (emit_aperture, points);
hp : pression en fonction du temps mesuré au niveau des points
start_time : instant correspondant aux premiers échantillons des vecteurs
             hp (en seconde)
emit_aperture : pointeur vers le transducteur concerné
points : vecteurs n*3 des coordonnées (x,y,z) des n points auxquels la
          pression est calculée
```

Mesurer les signaux ultrasonores au niveau d'une antenne

Ensemble des signaux reçus sur les transducteurs : fonction `calc_scat_multi`

```
>> [rf_data, tstart] = calc_scat_multi (emit_aperture, rec_aperture,
scat_position, scat_amplitude);
rf_data : matrice (Nechantillons*Nelements) des signaux reçus, un signal
          colonne par transducteur de l'antenne de réception
tstart : instant d'arrivée du premier écho, temps du premier échantillon
         de rf_data
emit_aperture : pointeur vers l'antenne émettrice
```

```
rec_aperture : pointeur vers l'antenne réceptrice
scat_position : coordonnées (x,y,z) des NbScatters diffuseurs
présents dans le milieu (matrice NbScatters*3)
scat_amplitude : amplitude des diffuseurs  (vecteur colonne NbScatters*1)
```