



a pas de multiples états, donc pas de risque de désynchronisation entre hôte et interface.

### 3. interpréteur de commande

S'il n'y a pas d'états, il n'y a pas non plus besoin d'un processus serveur tournant en permanence sur l'hôte. En pratique les requêtes de l'interface sont servies dans le cadre de l'interpréteur de commande habituel de l'hôte ("shell", interpréteur lisp, etc.). Un avantage est que l'interface de commandes de l'hôte demeure accessible à l'utilisateur (à travers une fenêtre de terminal incorporée à MapSignal). Un autre est que le protocole est robuste et facile à déboguer.

### 4. format des échanges

Les échanges d'information se font à l'aide de caractères ascii affichables (pas de caractères de contrôle, pas de bit de parité). Les requêtes et réponses utilisent un format lisible par l'homme (nombres en ascii), sauf pour les données graphiques.

### 5. code graphique "min-max"

Un signal est affiché sur l'écran du Mac sous la forme de son *enveloppe*. Celle-ci se matérialise graphiquement par un certain nombre de traits verticaux, à la résolution de l'écran. Ces traits sont définis par les ordonnées de leurs extrémités, qui dépendent simplement des *min* et *max* de groupes de *k* échantillons de signal (*k* est fonction de l'échelle horizontale):

$$a_n = \max_{i \in [kn, k(n+1)]} S_i$$

$$b_n = \min_{i \in [kn, k(n+1)]} S_i$$

Un algorithme spécial permet des échelles fractionnaires. Pour assurer la continuité visuelle (rompue lorsque  $a_n < b_{n+1}$  ou  $b_n > a_{n+1}$ ) ces valeurs sont remplacées avant affichage par:

$$A_n = \max\left(a_n, \frac{a_n + b_{n-1}}{2}, \frac{a_n + b_{n+1}}{2}\right)$$

$$B_n = \min\left(b_n, \frac{b_n + a_{n-1}}{2}, \frac{b_n + a_{n+1}}{2}\right)$$

Ces données graphiques sont calculées sur la machine hôte, codées sur 4 octets et envoyées à l'interface utilisateur qui les décode et s'en sert pour tracer l'enveloppe du signal. Le résultat graphique est identique à celui qui serait obtenu si tous les échantillons étaient dessinés, mais le volume de données à transmettre est beaucoup plus faible.

Ainsi, ce ne sont pas les données numériques qui transitent entre l'hôte et l'interface, mais seulement leur *représentation graphique*. Cela permet à MapSignal d'offrir un temps de réponse tout à fait honorable malgré la capacité limitée d'une liaison série. La gestion des fenêtres (réaffichage, etc.) est purement locale et ne nécessite aucun transfert de données.

### 3.3. Mécanisme d'extension

MapSignal n'offre en propre que des fonctions de visualisation, mais il possède un mécanisme qui permet de d'incorporer de nouvelles fonctions. Un "éditeur de commandes" permet de constituer un menu de commandes. A chacune d'elles correspond une chaîne de caractères qui est envoyée à l'hôte lorsque la commande est choisie dans le menu. La chaîne de caractères peut comporter des *mots-clé* qui sont interprétés au moment de l'envoi, ce qui permet d'inclure dans la chaîne des informations dynamiques telles que les coordonnées du curseur, le nom du fichier signal affiché, etc.. L'utilisateur peut ainsi diriger interactivement l'exécution de programmes disponibles sur l'hôte, en particulier ceux qu'il aura réalisés lui-même. Puisque l'affichage et l'interaction sont pris en charge par MapSignal, ces programmes peuvent être simples et portables.

### 3.4. Mécanisme d'interrogation de paramètres

Avant l'affichage d'un signal, MapSignal interroge l'hôte pour obtenir des informations sur ce signal (nombre d'échantillons, etc.). L'hôte envoie les informations qu'il possède: celles qui manquent sont remplacées par des valeurs par défaut. L'hôte a aussi le choix de la méthode par laquelle il obtient une information. Par exemple, la taille en échantillons du fichier signal peut provenir d'un "header", du système d'exploitation, ou d'un programme qui mesure explicitement la taille du fichier. MapSignal peut ainsi accommoder des formats de données très divers.

### 3.5. Moniteur d'acquisition

Une fonction originale mérite d'être signalée. Il s'agit d'une *fenêtre-moniteur d'acquisition de données analogiques*. Associée à un programme d'acquisition continue sur l'hôte, qui utilise un *buffer*

*circulaire* (buffer contenant en permanence les *n* dernières secondes de données), elle affiche son contenu. Ce contenu est en constant renouvellement au rythme de l'acquisition de nouvelles données, ce qui se reflète par l'image du signal qui "flotte" de droite à gauche à travers l'écran. L'utilisateur peut ainsi "geler" l'acquisition au moment opportun. On évite ainsi les problèmes de "timing" dus au mauvais alignement temporel, ou les distortions dues au mauvais réglage du niveau.

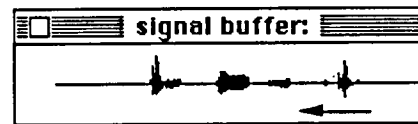


Fig. 3. Moniteur d'acquisition montrant le contenu d'un buffer circulaire.

L'acquisition continue est plus conviviale pour un locuteur que celle, discontinue, qu'imposent les solutions classiques.

### 3.6. Implémentation

MapSignal est écrit en C sur le Macintosh. Il nécessite deux commandes simples sur l'hôte, écrites elles aussi en C. Le programme est disponible gratuitement à fins de recherche. La distribution comprend le programme Macintosh, les sources des commandes hôte en C, et la documentation.

## 4. COMPARAISON AVEC X-11

L'idée principale défendue dans cet article est qu'on a intérêt à dissocier interface-utilisateur et traitement (calcul, entrées-sorties) et à les attribuer à des machines différentes. Cette idée sous-tend celle de "frontal" utilisée pour les bases de données ou le calcul scientifique (Mathematica). Elle se retrouve aussi dans X-windows, qui permet à un programme tournant sur une machine d'interagir indifféremment avec toute console du réseau. X-windows est de plus en plus répandu, mais il souffre des inconvénients suivants:

a) On ne peut associer que des machines possédant une version compatible de X-windows. Une mise à niveau partielle peut tout désorganiser, par ailleurs certaines machines de calcul n'offrent pas X. On perd ainsi dans la pratique une part

de la liberté de choix qu'est censée apporter la dissociation des fonctions.

b) X-windows est gourmand en ressources (mémoire, disque).

c) X-windows ne permet n'offre pas un environnement logiciel aussi complet que ceux disponibles sur les micro-ordinateurs (cette situation change avec l'apparition des émulateurs X pouvant s'intégrer dans l'environnement logiciel d'un micro).

En comparaison, le modèle de communication sur lequel est fondé MapSignal pose peu de conditions quant à la machine hôte ou la nature de la liaison. Il faut cependant reconnaître que les contraintes de protocole que sont la relation maître-esclave stricte et l'absence d'états impliquent une communication moins riche que celle offerte par X-windows.

## CONCLUSION

Le problème de la portabilité des logiciels de manipulation du signal peut trouver un élément de réponse dans la dissociation des fonctions de traitement du signal et d'interface utilisateur entre machines distinctes. La communication entre machines que nécessite cette solution peut se faire de façon simple et robuste dans le contexte de l'interpréteur de commande (shell) de la machine hôte. Conçu selon ces principes, le logiciel MapSignal permet la visualisation et la manipulation de données de signal se trouvant sur des machines très diverses. Sa conception fait appel à des idées originales qui peuvent être utiles dans d'autres contextes.

## REMERCIEMENTS

Une partie de ce travail à été effectué lorsque l'auteur était à *ATR Interpreting Telephony Research Laboratories* dans le cadre d'une bourse du programme STP des Communautés Européennes. Il tient à remercier ATR de son hospitalité.

## BIBLIOGRAPHIE

- [1] de Cheveigné, A., M. Abe and S. Doshita (1985), "The human interface of a speech workstation", *Studia Phonologica. XIX*, 18-26.
- [2] de Cheveigné, A. (1989), "The MapSignal remote speech editor", *Proc. Acoust. Soc. Japan Autumn meeting*, 383-384.
- [3] de Cheveigné, A. (1990), "The MapSignal remote speech editor", ATR Technical report TR-1-0137.