

Primeiro Trabalho Menor: Tolerância subjetiva a ruídos de banda-larga e banda-estreita

Data de entrega: 02/09/2014 até 12:00 pelo PACA

MUITA ATENÇÃO: OS TRABALHOS SÃO INDIVIDUAIS!

Tirar dúvidas “gerais” sobre o Pd no fórum é OK,
compartilhar implementações não é OK.

Neste trabalho nós testaremos empiricamente, através de um patch Pure Data, a validade da afirmação “*Fairly large amounts of broadband noise will often be tolerated more readily than fairly small amounts of narrowband noise*” (Moore, Elements of Computer Music, pg. 38). Criaremos um tocador de áudio que mistura ao arquivo tocado um sinal ruidoso de amplitude crescente, para estimarmos nosso limite de tolerância perceptual nos casos do ruído de banda-larga e banda-estreita.

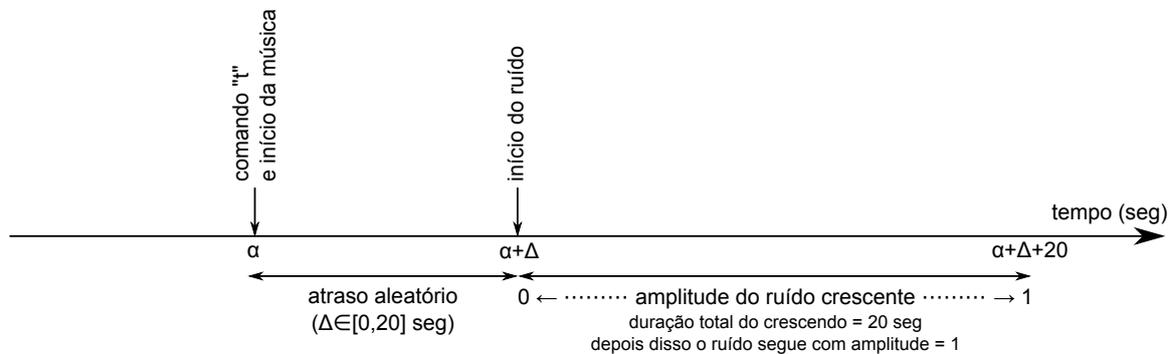
Para padronização, nossos patches deverão aceitar a seguinte lista de comandos do teclado:

a=abre: apresenta uma tela de seleção para escolher um arquivo de áudio

t=toca: inicia a reprodução do arquivo selecionado

p=para: para a reprodução e imprime o nível rms instantâneo do ruído naquele instante

A reprodução do arquivo de áudio deverá incorporar um mecanismo de introdução gradual do ruído; para que não sejamos influenciados pelo conhecimento do início do instante do ruído, ao iniciar a reprodução (comando “t”) o patch deve sortear um valor de espera entre 0 e 20 segundos, e só começar a introduzir o ruído a partir deste momento. Por convenção, a partir do momento em que entra o ruído, a amplitude deste deve subir quadraticamente entre os valores 0 e 1 num intervalo de 20 segundos:



A cada comando “t” nosso patch deverá sortear também o tipo de ruído que será aplicado naquela execução: banda-larga ou banda-estreita. Por convenção, usaremos como ruído banda-larga a saída do objeto¹ `[noise~]`, e como ruído de banda-estreita a conexão `[noise~]->[fexpr~ (0.14*$x1-0.99*$y[-2])]` (esta expressão concentra o ruído na faixa dos 11 kHz, mantendo a mesma amplitude rms do `[noise~]`). Ao interromper a execução com “p” nosso patch deverá imprimir, além do nível rms instantâneo do ruído, também o tipo de ruído que foi mixado à música naquela execução.

Depois de pronto, vocês devem testar o patch com a música² que está no link <http://tnb.art.br/download/?id=82536>. Mandem tocar 20 vezes a música, escutando o som gerado e interrompendo a execução assim que vocês perceberem a influência negativa do ruído. Tomem nota, para cada execução, do tipo de ruído e do nível em decibéis no momento em que vocês interromperam a execução. Enviem junto³ com o trabalho um pequeno texto contendo essas saídas e também as médias dos níveis em decibéis tolerados para cada tipo de ruído (uma média para banda-larga e outra para banda-estreita).

DICAS:

Alguns objetos serão úteis para essa implementação. Recomendo que vocês criem esses objetos e chamem o patch de ajuda respectivo (clique com o botão direito sobre o objeto): `[key]`, `[sel]`, `[rand]`, `[del]`, `[openpanel]`, `[readsf~]`, `[multiplex~]`, `[rms~]` e `[rmstodb]`. Para conectar os objetos `[openpanel]` e `[readsf~]`, uma solução prática é introduzir duas mensagens entre eles, contendo respectivamente `|set open $1, addcomma, add 1< e |<` (mensagem vazia); a primeira mensagem preencherá a segunda mensagem com o conteúdo requerido pelo `[readsf~]` para abrir o arquivo selecionado e tocar seu conteúdo.

Para controlar o volume do ruído, mandem uma mensagem `|0, 1 20000<` para um objeto `[line~]`, com a saída deste último conectada às duas entradas de um objeto `[*~]`. Isso elevará o valor da amplitude ao quadrado, para posteriormente modificar a amplitude do ruído (com outro `[*~]`) e finalmente somar (`[+~]`) ao sinal da música.

Finalmente, como uma sugestão, façam o desenvolvimento de forma incremental: testem separadamente a parte do patch que toca o arquivo, o sorteio do tipo de ruído e do instante de início, o controle de amplitude do ruído, a impressão dos resultados, etc. Vocês podem, conforme a preferência de vocês, fazer a implementação em um único patch, em um patch contendo alguns subpatches, ou ainda como vários arquivos .pd (abstrações) separados.

¹Os objetos Pd estão indicados entre [e], sendo que os colchetes não fazem parte do nome do objeto. Analogamente, as mensagens estão indicadas entre | e <, sendo que estes delimitadores também não fazem parte da mensagem.

²Sugerida na mensagem <http://paca.ime.usp.br/mod/forum/discuss.php?d=25709#p74096>. Observem que o objeto `[readsf~]` não aceita arquivos mp3, e portanto será necessário converter o arquivo baixado (por exemplo para wav). Essa conversão pode ser feita (por exemplo) com o comando `sox arquivo.mp3 arquivo.wav` (depende da instalação de `libsox-fmt-mp3`) ou `lame --decode arquivo.mp3 arquivo.wav`

³Façam um zip ou tgz ou rar. O PACA só aceitará o upload de um arquivo.