



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA



Escola d'Enginyeria de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Millora de la intel·ligibilitat de la paraula en llocs públics

Autor: Carlos Vega Carrera

Tutor: Andreu Balastegui

**PROJECTE FINAL DE GRAU**

Terrassa, 10 de Juny de 2015

<b>1. Introducció</b> .....	<b>4</b>
<b>1.1 Motivacions</b> .....	<b>4</b>
<b>1.2 Objectius del projecte</b> .....	<b>5</b>
<b>2. Fonaments d'acústica</b> .....	<b>6</b>
<b>2.1 So</b> .....	<b>6</b>
2.1.1 Ones sonores .....	7
2.1.2 Anàlisi espectral del so .....	8
2.1.3 Pressió sonora .....	9
2.1.4 Relació senyal-soroll.....	10
<b>2.2 L'oïda humana</b> .....	<b>10</b>
2.2.1 L'oïda externa .....	11
2.2.2 L'oïda mitjana .....	11
2.2.3 L'oïda interna .....	11
<b>2.3 Sonoritat</b> .....	<b>12</b>
2.3.1 Llindar d'audició .....	13
2.3.2 Corbes isofòniques .....	13
2.3.3 Emmascarament sonor.....	14
<b>2.4 La veu</b> .....	<b>17</b>
2.4.1 Sons fonats.....	18
2.4.2 Sons no fonats.....	19
<b>2.5 Acústica arquitectònica</b> .....	<b>19</b>
2.5.1 Ecos.....	19
2.5.2 Primeres reflexions .....	19
2.5.3 Absorció sonora .....	20
2.5.4 Temps de reverberació.....	20
<b>3. Intel·ligibilitat</b> .....	<b>22</b>
<b>3.1 Articulation Index (AI)</b> .....	<b>23</b>
<b>3.2 Speech Intelligibility Index (SII)</b> .....	<b>25</b>
<b>3.3 Speech Transmission Index (STI)</b> .....	<b>26</b>
3.3.1 Rapid Speech Transmission Index (RASTI).....	28
3.3.2 Speech Transmission Index for Public Address Systems (STIPA)....	28
<b>3.4 Short-Time Objective Intelligibility (STOI)</b> .....	<b>29</b>
<b>5. Implementació</b> .....	<b>31</b>

<b>5.1 Tasques preliminars a la implementació</b> .....	<b>31</b>
<b>5.2 Programari utilitzat</b> .....	<b>32</b>
<b>5.3 Disseny del programa</b> .....	<b>33</b>
5.3.1 Funcions dels índexs d'intel·ligibilitat.....	33
5.3.2 Funcions dels filtres .....	35
5.3.3 Codi principal .....	37
5.3.4 Funcions secundaries .....	41
<b>5.4 Millores proposades</b> .....	<b>43</b>
<b>5.4.1 Primera millora</b> .....	<b>43</b>
5.4.1 Segona millora .....	47
5.4.1 Tercera millora .....	47
<b>5.5 Funcionament del programa</b> .....	<b>49</b>
<b>6. Resultats</b> .....	<b>52</b>
<b>7. Conclusions</b> .....	<b>63</b>
<b>7.1 Futures implementacions</b> .....	<b>64</b>
<b>8. Bibliografia</b> .....	<b>65</b>
<b>9. ANNEX A - CODI</b> .....	<b>66</b>
<b>ANNEX A.1 – Funcions</b> .....	<b>66</b>
A.1.1 – Funcions pel càlcul de la intel·ligibilitat .....	66
A.1.2 - Funcions secundaries .....	84
<b>ANNEX A.2 – Codi GUI</b> .....	<b>89</b>

# 1. Introducció

La intel·ligibilitat de la paraula és un terme que s'utilitza per mesurar si el discurs d'una persona és comprensible pel receptor. Encara que sembla un terme científic i objectiu, realment no és així, ja que es mesura de forma subjectiva i són molts els factors que poden influir. Per aquest motiu, en aquest projecte avaluarem diferents tipus de índex que s'utilitzen per mesurar la intel·ligibilitat de la paraula i veurem en què es basa cadascun d'ells per tal de calcular de forma objectiva un terme tan subjectiu.

La intel·ligibilitat de la paraula es troba principalment a les freqüències d'entre 500 i 3500 Hz, mentre que a les freqüències més baixes trobem l'energia de la veu, i la seva eliminació faria que la veu tingués poca potència sonora.

En aquest projecte, com que no és possible aconseguir una disminució del soroll de fons, s'implementarà una forma de millorar la senyal de veu, per tal de millorar la intel·ligibilitat de la paraula.

## 1.1 Motivacions

En nombroses ocasions ens trobem amb llocs on la intel·ligibilitat de la paraula es molt important. Aquests llocs són normalment públics i podrien ser des d'una estació d'autobusos fins a un aeroport. En aquest tipus de llocs, moltes vegades és necessari entendre què diuen per tal de seguir les indicacions, però degut al soroll de fons que pot haver-hi, el nivell de la intel·ligibilitat no és suficient.

Una reducció del soroll de fons en aquests llocs seria ideal, però com això no és possible en la gran majoria dels casos, el què hem de fer és aplicar la millora a la senyal de veu per tal d'intentar aconseguir que el nivell de veu estigui per sobre del nivell de soroll i d'aquesta manera aconseguir que s'escolti millor el missatge que volem transmetre.

Un altre motiu a tenir en compte a l'hora de millorar la intel·ligibilitat de la paraula seria com mesurem aquesta millora. Com ja sabem després de llegir la introducció, aquest terme és molt subjectiu, i depenent de cada persona el resultat pot variar molt. És per això que durant el projecte veurem diferents

índexs per mesurar la intel·ligibilitat i com funcionen. A partir de estudiar-los traurem una conclusió de quins índexs funcionen d'una forma millor i quins funcionen d'una forma menys fiable.

## **1.2 Objectius del projecte**

Es tracta de fer un programa que sigui capaç de millorar la senyal de veu de forma que una vegada aquesta senyal ha estat tractada, millori la intel·ligibilitat de la paraula. Aquest programa, a més a més de tractar la senyal, serà capaç de donar un valor objectiu segons l'índex que elegim i tindrà un reproductor per tal de poder avaluar la millora d'una forma més subjectiva.

Aquest programa llegirà de forma separada la senyal de veu i la senyal de soroll, i el nivell senyal-soroll (SNR). A més a més d'aquest paràmetres, també tindrà uns paràmetres a ajustar per tal de calcular la millora. Per últim, s'haurà d'elegir entre els diferents índexs i es podran visualitzar els resultats de la millora.

El programa comptarà també amb un reproductor en el qual es podran escoltar tant la senyal sense millora com la senyal millorada, per tal de poder fer una comparació dels dos senyals.

## 2. Fonaments d'acústica

En aquest projecte no es pretén donar un curs d'acústica, però primerament hem de donar una visió global dels aspectes més importants per poder entendre tot el projecte. En aquest capítol es farà una introducció als aspectes més importants de l'acústica.

Denominem acústica a la branca de la física que estudia la producció, transmissió, percepció i reproducció del so. El so consisteix en una variació de la pressió en un medi elàstic, com per exemple l'aire o qualsevol matèria en qualsevol dels seus 3 estats. El so, com a ona mecànica, no és capaç de propagar-se en el buit.

### 2.1 So

El so es genera en crear-se una variació o pertorbació que estableix una sèrie d'ones de pressió (ones sonores) que fluctuen la pressió de l'aire i que es propaguen en totes les direccions des de la font sonora. Les partícules sotmeses a aquesta vibració no es desplacen, si no que oscil·len sobre el seu punt d'equilibri i transmeten l'oscil·lació a les partícules justament adjacents. La nostra oïda és capaç de transformar aquests canvis de pressió en impulsos elèctrics que es transmeten al cervell per a la seva posterior interpretació.

A l'hora d'analitzar un so existeixen tres elements bàsics a considerar:

- La font emissora, que pot ser desitjable, en el nostre cas la senyal de veu, o indesitjable, el soroll.
- El medi per el que es transmet el so.
- El receptor, en el nostre cas les persones que han d'escoltar el missatge.

Característiques del so:

- Intensitat: és la quantitat d'energia que transporta l'ona sonora per unitat de temps i de superfície en la direcció de propagació.
- To: ve associat a la freqüència de la component principal del so (harmònic fonamental).

- Timbre: està associat a la intensitat relativa d'altres freqüències a més a més de la freqüència fonamental.

### 2.1.1 Ones sonores

Les ones sonores s'originen per la creació d'un moviment oscil·latori. A l'oscil·lació hi ha dos moviments que s'expandeixen des del punt de producció:

1. De compressió degut a la pressió de l'aire
2. De rarefacció degut a l'absència de pressió

Al propagar-se el moviment oscil·latori, es genera una ona sinusoidal simple. Realment, les ones sonores són sempre el resultat de la adició d'un número determinat d'ones simples, aquestes s'anomenen ones compostes.

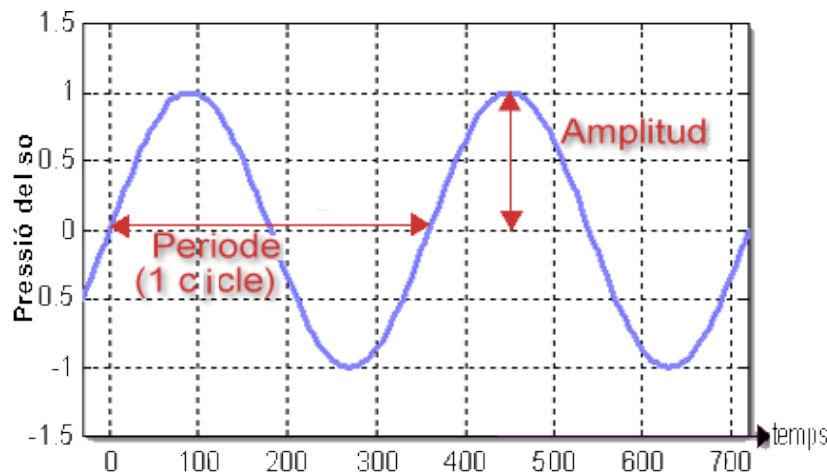


Figura 2.1.1 - Representació d'una ona sonora sinusoidal

Propietats de les ones sonores:

- Freqüència: número de cicles que completa en una unitat de temps l'ona sonora. Es mesura en Hertz (Hz), que són els cicles per segon.
- Període: l'invers de la freqüència, es tracta de la quantitat de temps que necessita per fer una oscil·lació. Es mesura en segons. A la figura 2.1.1 es pot veure la seva representació gràfica.
- Amplitud: distància des de la posició de repòs a la de màxima pressió. És una mesura de potència del moviment i és mesurada en decibels (dB). A la figura 2.1.1 es pot veure l'amplitud representada.
- Velocitat de propagació: depèn de les característiques del medi, pressió, temperatura, densitat i humitat. En general la velocitat de propagació és

major als sòlids que al líquids, i als líquids majors que al gasos. La velocitat del so a l'aire es de aproximadament 343 m/s, en condicions de temperatura de 20°C, humitat del 50% i a nivell del mar.

- Longitud d'ona: és la distància entre dos punts consecutius en el mateix estat de vibració en qualsevol instant de temps. Es mesura en metres i depèn de la velocitat de propagació i és inversament proporcional a la freqüència del so.

## 2.1.2 Anàlisi espectral del so

L'oïda humana es capaç d'escoltar freqüències entre 20 Hz i 20 kHz, però on es més sensible es a les freqüències al voltant d'1 kHz.

El cas més simple és un to pur, que conté una única freqüència. Qualsevol so periòdic pot representar-se per la superposició discreta de tons purs, on cadascun d'ells té la seva intensitat.

### 2.1.2.1 Bandes d'octava i de terç d'octava

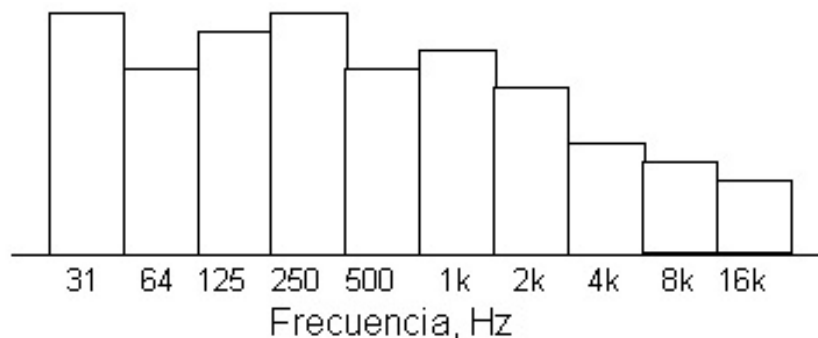


Figura 2.1.2.1 - Representació de les bandes d'octava

Excepte els tons purs o altres sons compostos d'un petit número de tons purs, la gran majoria dels sons estan compostos per una ampla varietat de freqüències. Per simplicitat, en comptes d'analitzar cada component de freqüències per separat, s'analitza en una sèrie d'interval de freqüències, cadascun d'ells caracteritzat per una certa freqüència, a la que se l'hi assigna tota l'energia corresponent a cada interval. A cada interval se'l denomina banda d'octava, on cada freqüència superior es el doble de la seva freqüència inferior. Les bandes d'octava es denominen pel valor de la freqüència central, que



bàsicament es tracta de la mitja geomètrica de les freqüències superior i inferior. L'amplada de banda és la diferència entre els límits de freqüència inferior i superior. A la *figura 2.1.2.1* es pot veure la representació d'un espectre en bandes d'octava.

Per mesures de major precisió, cada banda d'octava es divideix en tres bandes de terç d'octava, de forma que la relació entre les distintes freqüències és que la freqüència superior es igual a  $\sqrt[3]{2}$  vegades la freqüència inferior.

### 2.1.3 Pressió sonora

El so és una sensació auditiva provocada per les ones acústiques entre 20 Hz i 20000 Hz. Quan les freqüències són inferiors a 20 Hz es consideren infrasons i quan són superiors a 20000 Hz es consideren ultrasons. La magnitud que normalment es mesura amb un micròfon és la pressió eficaç.

Nivell de pressió sonora (dB)	
180 dB	L'explosió del volcà Krakatoa
140 dB	Llindar del dolor
130 dB	Avió enlairant-se
120 dB	Motor d'avió en marxa
110 dB	Concert
100 dB	Trepant elèctric
90 dB	Trànsit vehicular / Baralla
80 dB	Tren
70 dB	Aspiradora
50/60 dB	Aglomeració de gent
40 dB	Conversa
20 dB	Biblioteca
10 dB	Respiració tranquil·la
0 dB	Llindar d'audició

Taula 2.1.3 - Exemples de nivells de pressió sonora

Les diferències de pressió acústica són molt grans, existeixen variacions de pressió que varien entre 0,00002 i 200 Pa. Per aquest fenomen, s'ha introduït com a magnitud pràctica el nivell de pressió sonora en decibels (dB). Aquesta es calcula respecte d'una pressió de referència i amb escala logarítmica, de tal forma, es simplifica la forma de treballar amb la pressió.

$$\text{SPL(dB)} = 20 \log \left( \frac{P_{ef}}{P_{ref}} \right) \text{ on } P_{ref} = 0,00002 \text{ Pa}$$

A la taula 2.1.3 es possible veure exemples de nivells de pressió sonora, per fer-nos una petita idea del que estem parlant.

### 2.1.4 Relació senyal-soroll

La relació senyal-soroll, també coneguda com a SNR, és defineix com la proporció existent entre la potència de senyal que es transmet i la potència del soroll. Aquest marge està normalment mesurat en decibels (dB).

## 2.2 L'oïda humana

L'oïda humana és capaç de detectar sons en un interval de 20 Hz a 20000 Hz i és a 1000 Hz on és capaç de detectar sons amb una amplitud més baixa. Però, l'oïda no és només una receptor de banda ampla, combinat amb el sistema nerviós, es tracta d'un analitzador d'espectre amb una gran selectivitat. A la figura 2.2 podem veure una il·lustració de l'oïda humana amb les seves diferents parts.

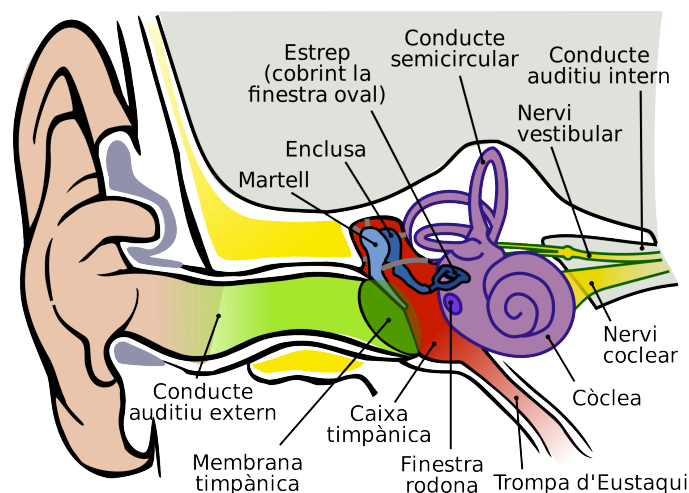


Figura 2.2 - Oïda humana

L'oïda humana consta de tres parts principals: l'oïda externa, mitjana i interna.

### **2.2.1 L'oïda externa**

L'orella capta el so i l'envia cap al canal auditiu. És la part més externa i l'única visible.

El canal auditiu, és un tub aproximadament recte de 0,7 cm de diàmetre i 2,5 cm de llarg, tancat en el seu extrem interior pel timpà. Aquest tub condueix el so cap al timpà, i a més a més actua com a ressonador en freqüències entre 2 kHz i 5 KHz.

El timpà és una membrana elàstica i en forma de con aplanat capaç de transformar la pressió sonora en vibracions elèctriques que es transmeten a l'oïda mitjana.

### **2.2.2 L'oïda mitjana**

És una cavitat plena d'aire d'aproximadament 2 cm de volum que contè 3 ossicles.

Els ossicles són una cadena de 3 ossets: martell, enclusa i estrep, que juntament amb els músculs que els sostenen tenen la funció de l'amplificació del nivell de pressió sonora que arriba al timpà. Aquesta cavitat està connectada amb la trompa d'Eustaqüi.

La trompa d'Eustaqüi és un conducte que té la funció de de fer que es mantingui l'equilibri entre les pressions externes i internes.

### **2.2.3 L'oïda interna**

També coneguda com el laberint, consta de tres parts: el vestíbul, els conductes semicirculars i la còclea.

El vestíbul es connecta amb l'oïda mitjana a través de dues obertures, la finestra oval i la finestra circular. Ambdues finestres estan tancades per tal que no s'escapi el líquid que hi ha dins de l'oïda interna.

Els conductes semicirculars no tenen cap funció en el procés de l'audició, però ens proporcionen el sentit de l'equilibri.

La còclea es un tub circular enrotllat en forma de cargol, que està dividit per la partició coclear en dos canals longitudinals diferents, el vestibul superior i el vestibul inferior.

El nervi auditiu consta de dues branques diferenciades, el nervi vestibular i el nervi coclear. El nervi vestibular és el que porta la informació del sentit del equilibri. Mentre que el coclear està connectat amb totes les cèl·lules ciliades que porten tota la informació sonora.

## 2.3 Sonoritat

La sonoritat es defineix com aquella propietat de la sensació auditiva que permet ordenar els sons en una escala que va des dels fluixos fins els forts. Per tant, es tracta d'una experiència subjectiva, diferent per cada individu.

Per quantificar la relació entre intensitat sonora i sonoritat, es va definir una unitat de sonoritat, anomenada com el "son", fent que 1 son correspongués a la sonoritat d'un to sinusoidal d'1 kHz a 40 dB SPL en camp lliure.

La sonoritat no depèn només de la intensitat, amb tons purs de la mateixa intensitat es pot comprovar que a diferents freqüències la sensació de sonoritat és diferent, per exemple, un so molt agut generarà molta menys sonoritat que no pas un to a una freqüència mitjana.

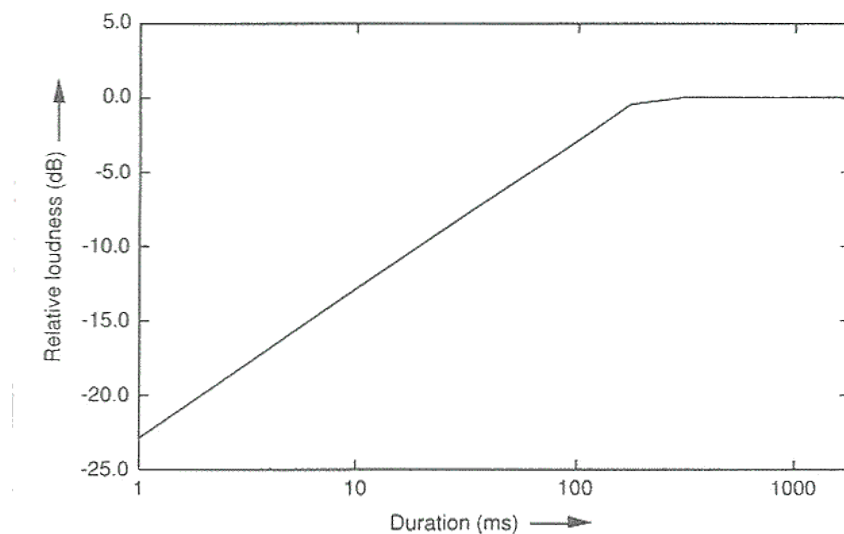


Figura 2.3 - Sensació de sonoritat respecte a la durada del so

La sonoritat també depèn de la durada del so, d'aquesta manera, per a sons amb una durada de menys de 150 ms i la mateixa intensitat, cada vegada que el temps creix en un factor de 10, la sonoritat augmenta 10 dB. Aquest fenomen succeeix fins a una durada de 200 ms, a partir que el so té una durada de més de 200 ms, ja no es tradueix en un augment de sonoritat. Això es pot observar representat a la *figura 2.3*.

### 2.3.1 Llindar d'audició

El llindar d'audició és el mínim nivell de pressió sonora necessari per produir una sensació de sonoritat. Aquest nivell és normalment expressat en decibels. Aquest límit serà diferent per a cada freqüència i els valors exactes que son utilitzats venen a la norma ISO 389-1.

### 2.3.2 Corbes isofòniques

Una vegada hem vist que la sonoritat depèn tant de la intensitat com de la freqüència, per mesurar qualsevol so, s'utilitzen les corbes isofòniques, on cada corba porta associada una xifra que indica el nivell de sonoritat corresponent en tot l'espectre audible. Aquestes corbes van relacionades amb la capacitat que té l'oïda humana per captar els sons, per tant, a baixa freqüència es necessita més nivell de potencia per adquirir la mateixa sonoritat que a freqüències entre 1 i 5 kHz, on l'oïda és més sensible. A la *figura 2.3.2* es pot veure la representació de les diferents corbes isofòniques, on podem observar la diferència de sonoritat en intensitat i freqüència.

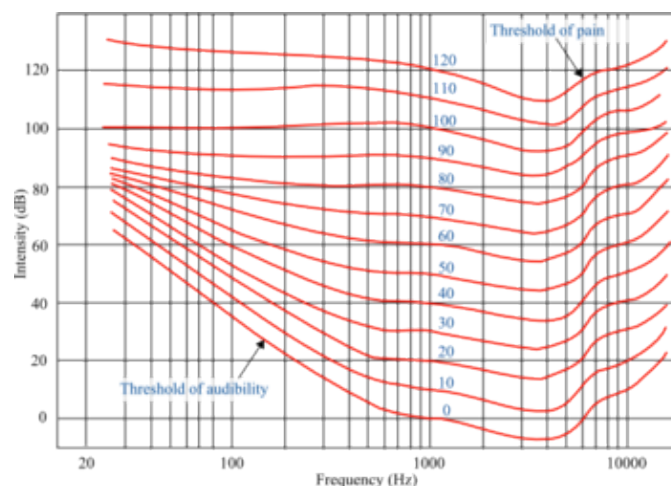


Figura 2.3.2 - Representació de les corbes isofòniques

### 2.3.3 Emmascarament sonor

L'emascarament sonor pot definir-se com el procés en el qual el llindar d'audició corresponent a un so s'eleva, degut a la presència d'altre so. Per exemple, si s'escolta una senyal sonora sempre al mateix nivell i poc a poc es va augmentat progressivament el nivell d'un soroll de fons, mentre es manté constant la senyal sonora inicial, la sensació serà de què cada vegada que el soroll de fons es més alt, la intensitat del senyal sonor va disminuint fins que es fa inaudible.

L'emascarament depèn del nivell de pressió sonora de les senyals "emascarant" i "emascarada" a més a més de la separació en freqüència i temps de les mateixes.

Es defineix com nivell de sensació (NS) d'una senyal com la diferència, en dB SL, entre el llindar d'emascarament (LE) i el llindar d'audició (LA) corresponents a la senyal i expressats en dB SPL.

$$NS_{dB_{SL}} = LE_{dB_{SPL}} - LA_{dB_{SPL}}$$

Depenent de la ubicació temporal de la senyal sonora (E) respecte de la senyal emascarant (P) es poden distingir tres situacions diferents.

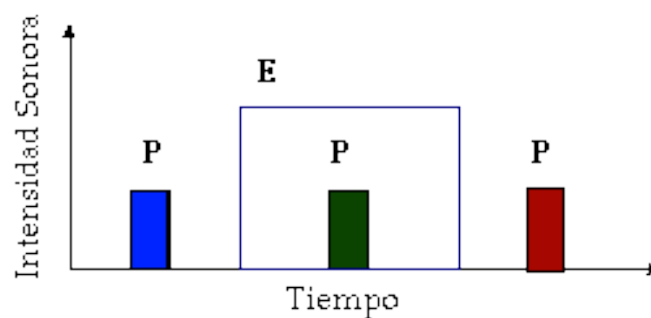
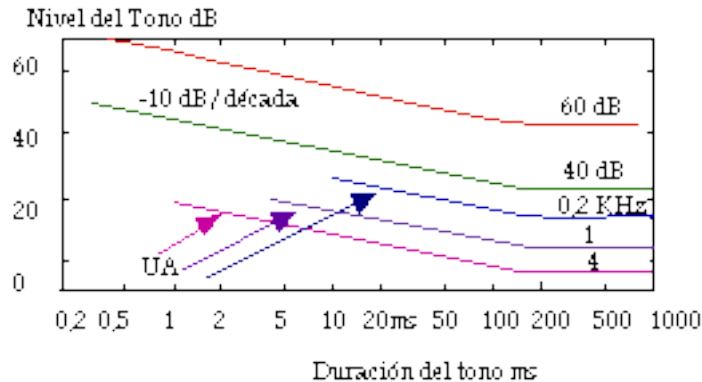


Figura 2.3.3.1 - Emmascarament temporal

- Emmascarament simultani: E i P es presenten solapats en el temps.
- Emmascarament previ a la presentació de la senyal emascarant, E es presenta després de P.
- Emmascarament posterior a la presentació de la senyal emascarant, E es presenta abans que P.

A la *figura 2.3.3.1* es pot observar el fenomen de l'emascament temporal.

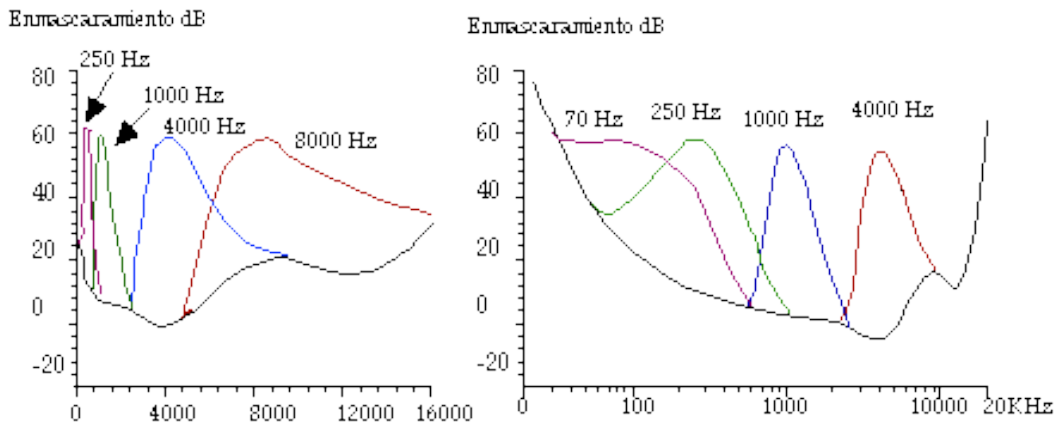
Tant el llindar d'audició com el llindar d'emascament depenen de la duració de la senyal sonora. La següent gràfica mostra la variació del llindar d'audibilitat per tons de 200, 1000 i 4000 Hz, així com la variació del llindar d'emascament produït pel soroll uniformement emmascarat (RUE), en funció de la duració de la senyal emmascarada.



**Figura 2.3.3.2 - Emmascament dependent de la duració de la senyal sonora**

A la *figura 2.3.3.2* es demostra que per duracions inferiors a 200 ms, el sistema auditiu treballa com un detector d'energia. Per altre banda, per duracions de més de 200 ms, el llindar es independent de la duració, en poques paraules, es pot dir que el sistema auditiu treballa com un detector d'energia amb una finestra de 200 ms de duració.

En la següent figura es pot observar el patró d'emascament provocat per bandes de soroll d'una banda crítica d'amplada de banda i de nivell de 60 dB SPL. Les bandes de soroll estan centrades en 70, 250, 1000, 4000 i 8000 Hz i tenen amplades de banda de 100, 100, 160, 700 i 1700 Hz, respectivament. Les pendents superior i inferior de cada banda de soroll són superiors a 200 dB/octava, per tant, son espectres considerats rectangulars. La corba inferior correspon al llindar d'audició.



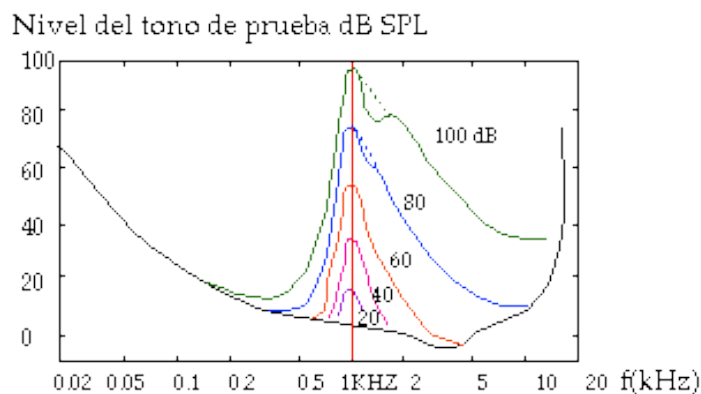
**Figura 2.3.3.3 - Emmascarament freqüencial**

L'efecte d'emascarament s'estén fora de l'interval de freqüències en el qual esta la senyal emmascarant, també podem observar com s'estén més per a les senyal d'alta freqüència que no pas per les de baixa freqüència.

El valor del llindar d'emascarament baixa al augmentar la freqüència central de la banda de soroll. Tot i que tots els tons d'emascarament son de 60 dB SPL, podem veure com a la banda de 250 Hz el llindar d'emascarament es de 58 dB SPL, mentre que a la banda de 4000 Hz el llindar d'emascarament es de 55 dB SPL.

Aquest dos efectes mencionats anteriorment es poden observar a la *figura 2.3.3.3*, on es veu un exemple de l'emascarament freqüencial.

A la *figura 2.3.3.4*, també es pot observar que el patró d'emascarament depèn del nivell de la senyal emmascarant.



**Figura 2.3.3.4 - Efecte emmascarament a diferents intensitats**



## 2.4 La veu

El nivell mig total de la veu a una distància d'1 metre és d'aproximadament 65 dB. La veu masculina té un to fonamental d'entre 100 i 200 Hz, mentre que la veu femenina es més aguda, típicament entre 150 i 300 Hz. I les veus infantils són encara més agudes. Un altre característica important de la veu humana és que té un ampli espectre que va aproximadament des de 20 Hz fins a 20000 Hz. Però, per aconseguir una veu comprensible no és necessari transmetre totes les freqüències, sinó que amb un rang molt més petit ja podem aconseguir-ho. Per aquest fet, els telèfons comercials només transmeten en un rang aproximat de 400 a 4000 Hz. El rang aproximat de la veu humana es pot observar a la *figura 2.4*.



Figura 2.4 - Rang de la veu humana

L'energia acústica associada amb el llenguatge s'origina als músculs pectorals, que al contreure's, fan que l'aire dels pulmons passi a través de diverses components del mecanisme vocal. Aquest flux constant d'aire es pot veure com el portador d'energia que ha de ser modulats en velocitat i pressió per produir el so. La modulació requerida s'obté de dues formes fonamentals que es porten respectivament els sons fonats o no fonats.

A la *figura 2.4.1* es pot veure l'aparell fonador amb el que s'origina el llenguatge.

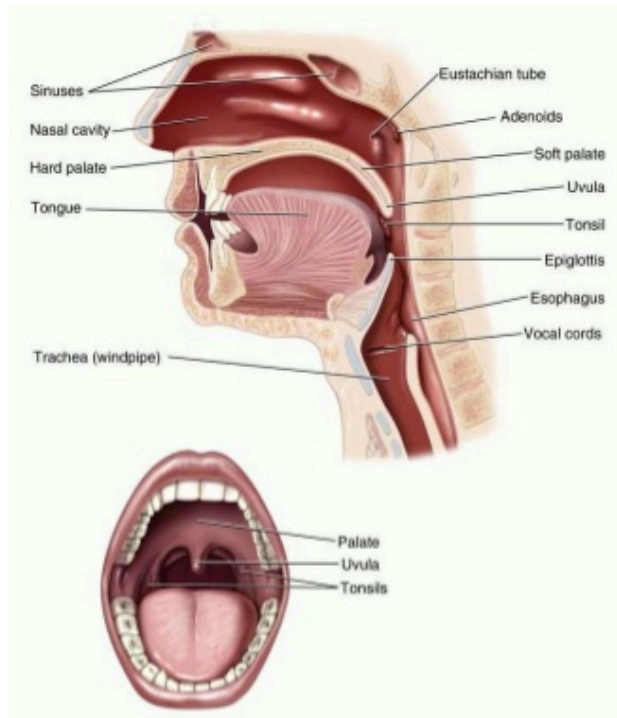


Figura 2.4.1 - Aparell fonador

## 2.4.1 Sons fonats

Els sons fonats porten les vocals, així com els tons de la veu constant. L'agent primari en la modulació és la laringe, a través de la qual s'estenen les cordes vocals.

Les cordes vocals són dues bandes semblants a membranes que formen un diafragma amb una obertura que modula la corrent d'aire en obrir-se i tancar-se conforme vibra. La longitud d'aquesta membrana és d'aproximadament 2,5 cm als homes i de 1,5 cm a les dones, i la tensió amb què les cordes vocals estan estirades, determinen la freqüència fonamental d'aquesta modulació. Les cordes vocals de les dones són més curtes i lleugeres, així que vibren a una freqüència de quasi el doble respecte dels homes, cosa que explica la major altura de la majoria de les veus de les dones. L'acció de les cordes vocals produeix una pressió en forma de dent de serra que conté molts harmònics.

Les cavitats ressonants i orificis del nas, boca i gola formen una xarxa de filtrat acústica que altera les quantitats relatives d'harmònics. Canviant la posició de

la llengua o la configuració dels llavis es poden produir una gran quantitat de sons diferents.

## 2.4.2 Sons no fonats

El mecanisme de veu, també és capaç de produir sons no fonats. Aquests sons inclouen les consonants no fonades com la f i la s, així com les consonants explosives p, t i k. En aquest cas, els sons es produeixen modulant la corrent d'aire amb els llavis, dents o llengua. Com en el cas dels sons fonats, el control de la llengua i llavis altera les ressonàncies i fa que siguem capaços de generar una gran varietat de sons no fonats, de tal manera que es pot aconseguir un llenguatge recognoscible, anomenat murmur.

## 2.5 Acústica arquitectònica

Un altre aspecte que és necessari tenir en compte quan estudiem la intel·ligibilitat de la paraula, és l'acústica de la sala on la veu serà escoltada, sempre que aquesta sigui escoltada en una sala en comptes d'en camp lliure.

L'acústica arquitectònica estudia els fenòmens vinculats amb la propagació adequada, fidel i funcional del so en un recinte, sigui quin sigui el recinte. Cada recinte tindrà una sèrie de propietats relacionades amb el comportament del so en el recinte, entre les quals es troben les primeres reflexions, la reverberació, l'existència o no d'ecos i ressonàncies, etc.

### 2.5.1 Ecos

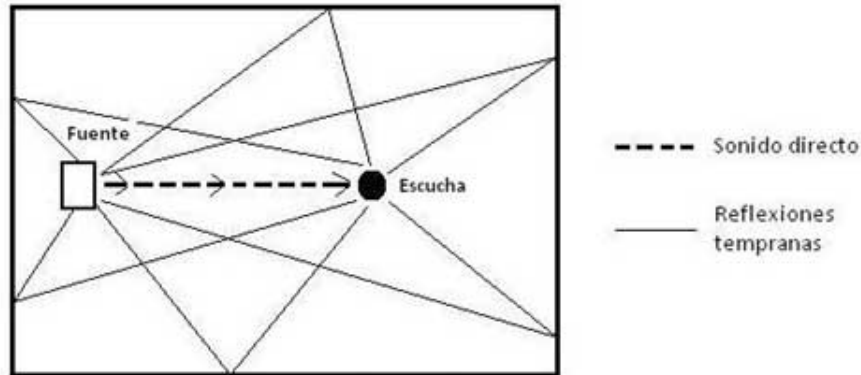
És el fenomen més senzill que apareix en un recinte amb superfícies reflectores de so. L'eco consisteix en una única reflexió que retorna al punt on es troba la font uns 100 ms o més després d'emetre el so.

$$t = \frac{2d}{c}$$

### 2.5.2 Primeres reflexions

Quan una font sonora està envoltada de diferents superfícies, un receptor rebrà el so directe i a més a més, el so reflectit en cada superfície. Les primeres

reflexions rebudes, que es troben bastant separades en temps, s'anomenen primeres reflexions. En sales petites, les diferències de temps són petites, i per tant, no s'arriben a percebre com eco. A la *figura 2.5.2* s'il·lustra una representació del so directe i les primeres reflexions en una sala.



**Figura 2.5.2 - So directe i primeres reflexions**

### 2.5.3 Absorció sonora

Les superfícies d'un recinte reflecteixen només parcialment el so que incideix, la resta del so és absorbit. Segons el material o recobriment de la superfície, aquesta podrà absorbir més o menys so.

El coeficient d'absorció sonora és el coeficient entre l'energia absorbida i l'energia incident.

$$\alpha = \frac{E_{\text{absorbida}}}{E_{\text{incident}}}$$

### 2.5.4 Temps de reverberació

Després de les primeres reflexions, comencen a aparèixer les reflexions d'aquestes primeres reflexions, i després les reflexions d'aquestes i així successivament. Aquesta permanència de so encara després de ser interrompuda la font sonora es denomina reverberació.

Ara bé, en cada reflexió, una part és absorbida per la superfície. Aquesta part absorbida pot transformar-se en minúscules quantitats de calor, propagar-se a un altre recinte veí o ambdues coses. La part reflectida, viatjarà fins a trobar-se

amb un altre superfície, i es tornarà a repetir aquest procés fins que el so reflectit ja sigui massa dèbil per ser audible i desapareix.

Per mesurar aquest procés d'extinció del so, s'introdueix el concepte de temps de reverberació, tècnicament definit com el temps que demora un so en baixar 60 dB respecte al nivell inicial.

El temps de reverberació depèn de la quantitat d'absorbents siguin les superfícies del recinte, així que si les parets són molt reflectants el temps de reverberació serà molt alt, en canvi, si les parets són molt absorbents, el temps de reverberació serà molt petit.

Aquesta propietat anterior es pot expressar mitjançant la fórmula de Sabine, que no es l'única forma de calcular el temps de reverberació, però si la més senzilla i ràpida d'aplicar.

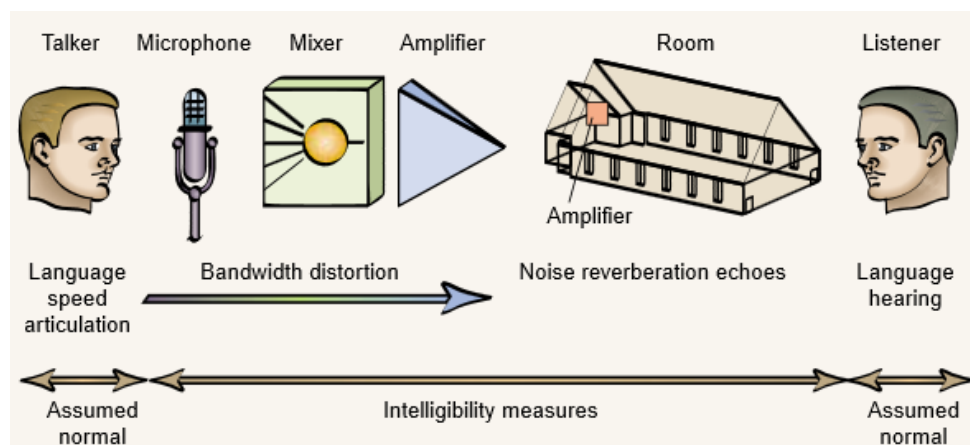
$$RT_{60} = \frac{V}{\sum \alpha \cdot S}$$

### 3. Intel·ligibilitat

En algunes situacions, com ara, grans edificis, grans indústries, aeroports o estacions de tren, la veu humana és utilitzada com a font d'informació pels receptors. Un missatge que no és "comprensible" podria no ser interpretat de la forma correcta, i això pot resultar en errors per part del receptor.

Hi ha moltes causes que poden portar a una mala interpretació per part del receptor, per exemple, una persona parlant ràpidament o en una llengua diferent a la del receptor. Però aquests són els casos més subjectius i que depenen més de l'ésser humà, això no es pot mesurar objectivament. Sense cap mena de dubte, un dels majors problemes que trobem a l'hora d'entendre un missatge és el soroll de fons que pot causar una mala interpretació del missatge, i és amb això amb el que basarem la mesura de la intel·ligibilitat de la paraula.

La *figura 3* és útil per tal d'entendre el camí que ha de seguir la veu des que s'envia per part de l'emissor, fins que la rep el receptor.



**Figura 3 – Camí de la senyal de veu**

A la *figura 3* es pot observar els problemes que poden ser introduïts al missatge durant les diferents etapes que ha de superar el senyal. L'error es va sumant successivament a cada etapa, i això fa que pugui afectar a la interpretació del mateix. Per exemple, un missatge es pot entendre quan l'emissor i el receptor es troben cara a cara, però el sistema de comunicació pot adherir distorsió, o el receptor es pot trobar en un lloc sorollós, que resultarà en què augmentin les possibilitats de que no sigui capaç d'entendre el missatge degudament.

Per què la paraula sigui intel·ligible, hem de tenir en compte dos factors, una adequada audició i una bona claredat de veu. L'audició ve relacionada amb la relació senyal-soroll (SNR), d'aquesta manera, a més SNR tinguem l'audició del senyal serà millor. Per altra part, la claredat és la propietat del so que permet que els fonemes siguin distingits de forma correcta pel receptor. La claredat d'un so pot ser reduïda per l'amplitud de la distorsió causada pels sistemes electrònics, la distorsió en freqüència per aquests mateixos sistemes i el temps de reverberació, a més a més del possible soroll de fons que es tingui on es troba el receptor.

En resum, la intel·ligibilitat de la paraula és la mesura de l'efectivitat amb la qual un discurs serà interpretat. La mesura és normalment expressada com el percentatge del missatge que ha estat entès correctament. La intel·ligibilitat de la paraula no implica qualitat del discurs, per exemple, una veu sintetitzada pot ser completament entesa, però en canvi, la qualitat d'aquesta veu pot ser molt baixa, es a dir, un missatge que no té una bona qualitat pot ser perfectament entès.

### **3.1 Articulation Index (AI)**

*L'Articulation Index* ens dona una mesura de la intel·ligibilitat de la paraula donat un ambient sorollós. El mètode va ser primerament desenvolupat l'any 1949 per tal de poder donar un valor a la intel·ligibilitat d'un sistema de comunicació. La interpretació del valor d'aquest índex es molt senzilla, al valor va en un rang de 0 a 1, on els valors més propers a 1 son la part més intel·ligible i els valors més propers a 0, els que donen una intel·ligibilitat més baixa.

El mètode per avaluar l'AI és bastant senzill, es tracta bàsicament de mesurar les diferències entre el senyal i el soroll de fons particular dividint l'espectre de les dos senyals en 20 bandes, les quals es poden veure a la *taula 3.1*, on totes aquestes bandes contribueixen de forma equitativa. Si el nivell del soroll de fons a la banda que estem analitzant és superior a la del nivell de senyal, la contribució a l'AI serà de 0. En canvi, si la senyal d'entrada té un nivell superior al soroll de fons, hi haurà una contribució. En el cas que el nivell de la

contribució sigui superior a 30 dB, llavors la contribució per aquesta banda serà de 30 dB com a màxim. Finalment, es sumaran les contribucions de totes les bandes i el valor es dividirà per 600. El valor resultant d'aquesta operació serà el valor final de l'AI.

Número de banda	Límits	Freqüència central
1	200 - 330 Hz	270 Hz
2	330 - 430 Hz	380 Hz
3	430 - 560 Hz	490 Hz
4	560 - 700 Hz	630 Hz
5	700 - 840 Hz	770 Hz
6	840 - 1000 Hz	920 Hz
7	1000 - 1150 Hz	1070 Hz
8	1150 - 1310 Hz	1230 Hz
9	1310 - 1480 Hz	1400 Hz
10	1480 - 1660 Hz	1570 Hz
11	1660 - 1830 Hz	1740 Hz
12	1830 - 2020 Hz	1920 Hz
13	2020 - 2240 Hz	2130 Hz
14	2240 - 2500 Hz	2370 Hz
15	2500 - 2820 Hz	2660 Hz
16	2820 - 3200 Hz	3000 Hz
17	3200 - 3650 Hz	3400 Hz
18	3650 - 4250 Hz	3950 Hz
19	4250 - 5050 Hz	4650 Hz
20	5050 - 6100 Hz	5600 Hz

Taula 3.1 – 20 bandes de contribució equitativa per a la intel·ligibilitat de la paraula

A la figura 3.1 podem trobar una aproximació de la relació que hi ha entre el valor de l'AI i la intel·ligibilitat de la paraula.

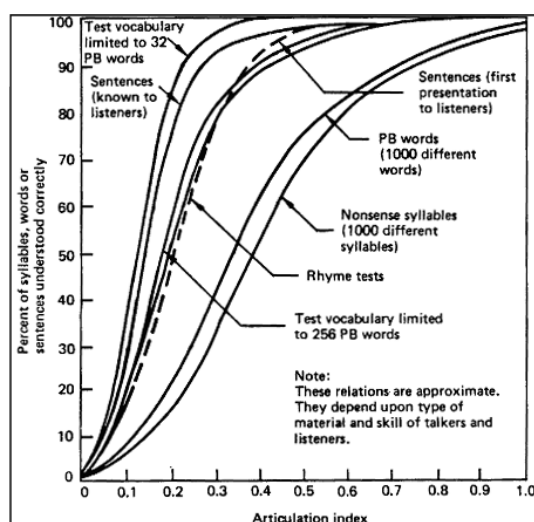


Figura 3.1 - Relació entre el AI i diferents mesures de la intel·ligibilitat de la paraula



## 3.2 Speech Intelligibility Index (SII)

Després que l'AI fos creat i utilitzat, molts investigadors van començar a intentar trobar un índex que funcionés d'una forma més fiable. Amb la re-examinació de l'*Articulation Index*, a l'any 1997 es va acceptar un nou mètode acceptat com a *ANSI S3.5-1997*, aquest nou mètode és anomenat *Speech Intelligibility Index* (SII).

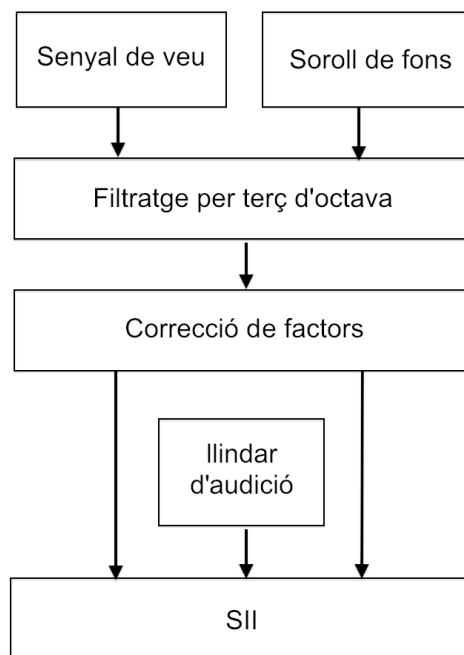
Per un senyal de veu donat amb el seu soroll de fons, el SII es calcula a partir de l'espectre del senyal de veu, l'espectre del soroll i el llindar d'audició de l'oient. Ambdós, senyal i soroll, són filtrats dins d'unes bandes, on no totes les bandes tenen la mateixa importància pel càlcul de la intel·ligibilitat. La importància de cada banda indica quina és la quantitat de contribució que té cada banda a la intel·ligibilitat de la paraula.

Finalment, el valor del SII ve determinat per l'acumulació d'audició durant les diferents bandes, ponderant cada freqüència per la importància de cadascuna de les bandes. El valor resultant és un valor entre 0 i 1, i aquest valor pot ser interpretat com la proporció total de informació disponible pel receptor. El SII ha estat desenvolupat per tal de predir una mitja de la intel·ligibilitat d'una senyal de veu donada amb el seu soroll de fons corresponent, per tant, no és capaç de predir la intel·ligibilitat de parts individuals del senyal (paraules o fonemes).

El model de SII només ha estat validat per sorolls de fons estacionaris. En canvi, si volem predir la intel·ligibilitat de forma acurada amb sorolls de fons molt canviants, el mètode SII no seria una bona opció. Altres mètodes com el STI o l'AI també tenen problemes per predir senyals amb aquest tipus de soroll de fons, de fet, encara no hi ha un mètode que sigui capaç de predir de forma suficientment acurada aquests tipus de casos.

El model SII bàsicament calcula la mitjana de la quantitat d'informació del senyal de veu que és comprensible per l'oient. Per aconseguir això, el model utilitza l'espectre del senyal de veu i l'espectre de la senyal de soroll de fons com a entrades. Ambdós espectres són definits com a nivell espectral, donat en dB/Hz, a l'orella del receptor. Dintre del model hi ha diferents opcions en el tractament de les bandes de freqüència, durant aquest projecte s'ha utilitzat el

model que utilitza 18 bandes definides dins del estàndard ANSI S3.5-1997. A cada banda de freqüència tindrem dos nivells espectrals, un pel senyal de veu i un altre pel soroll de fons. Després d'això, s'aplica la correcció d'alguns factors, com és el cas de l'emascament, el llindar d'audició del receptor i la distorsió a causa d'un excessiu nivell d'alguna dels senyals. Una vegada s'han fet totes aquestes correccions, la diferència entre el senyal d'entrada i el soroll de fons (SNR) és calculat i el seu valor per cada banda és ponderat per la importància de la banda corresponent, que resulta de la proporció d'informació que té cada banda. Es pot veure el camí que es segueix per fer el càlcul gràficament a la *figura 3.2*. Finalment, aquests valors son sumats i donen el valor del SII.



**Figura 3.2 - Camí per el càlcul del SII**

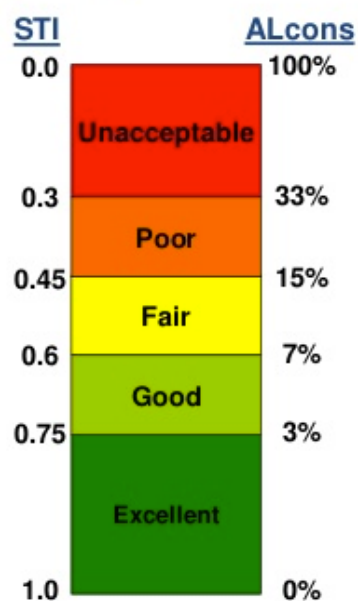
### 3.3 Speech Transmission Index (STI)

El Speech Transmission Index (STI) és un mètode per mesurar la intel·ligibilitat de la paraula en ambients sorollosos o reverberants. El STI i el seu germà petit, Rapid Speech Transmission Index (RASTI) són calculats a partir d'una resposta impulsiva, modulation transfer function (MTF). La MTF esta definida com la magnitud de la transformada de Fourier de l'arrel quadrada de la resposta impulsiva dividida per l'energia total de la resposta impulsiva. Els mètodes STI i

RASTI es mesuren en una escala que va des de 0 fins a 1, a la *figura 3.3* es pot observar la seva escala respecte a ALcons.

El ALcons és un altre mètode per mesurar la intel·ligibilitat de la paraula, aquesta mesura expressa la pèrdua de la definició de les consonants, on els valors petits indiquen una bona intel·ligibilitat. Es compara amb aquest mètode, ja que el ALcons és un mètode àmpliament utilitzat per molts acústics, especialment als Estats Units, tot i que té grans limitacions que venen donades per la forma amb la que es calcula. Aquest mètode esta basat en una sola banda de terç d'octava centrada a 2 KHz, per tant, la resta de l'espectre freqüencial es ignora, cosa que el fa no ser un mètode del tot fiable.

Al test STI, no s'utilitza la veu pròpiament dita, sinó que s'utilitza una senyal amb característiques similars a la veu humana. Aquest concepte es basa en què la veu pot ser representada com una forma d'ona fonamental que és modulada per senyals de baixa freqüència. El mètode pel càlcul de l'índex STI utilitza un esquema molt complex de modulació de l'amplitud per tal de generar aquesta senyal de prova. A la posició del receptor, la profunditat de la modulació que es rep és comparada amb el senyal original en diverses freqüències, les reduccions en la profunditat de la modulació són la pèrdua de la intel·ligibilitat.



**Figura 3.3 - Mesura del STI/RASTI respecte ALCons (percentatge de consonants perdudes)**

### **3.3.1 Rapid Speech Transmission Index (RASTI)**

Donada la complexitat que suposa el càlcul del STI, es va crear el seu germà petit denominat com RASTI. A diferència que el STI, aquest és molt més simple i només fa mesures a dues octaves, una centrada a 500 Hz i una altra centrada a 2000 Hz. De la mateixa manera que el STI, també utilitza un senyal que emula la veu humana i associa la pèrdua de la profunditat de la modulació a una reducció de la intel·ligibilitat.

Aquest mètode només avalua dues bandes de freqüència, donant per fet que el sistema s'estén d'una forma més o menys plana en tot l'espectre de la veu. Aquest és un gran inconvenient, ja que molts reproductors no són capaços de reproduir la veu en un espectre pla. D'aquesta manera un sistema que només fos capaç de reproduir en aquestes dues freqüències on s'avalua, obtindria una valoració perfecta tot i que podria emetre un senyal de veu totalment incomprensible pel receptor.

### **3.3.2 Speech Transmission Index for Public Address Systems (STIPA)**

STIPA és una altre versió simplificada del STI, aquest mètode va ser dissenyat per tal de fer mesures de la intel·ligibilitat en situacions específiques, com ara, sistemes de reproducció a locals públics, com ara aeroports o estacions de tren.

Com en els altres dos casos, s'utilitza un senyal de test que intenta simular les característiques d'un senyal de veu. Aquest mètode té uns càlculs més acurats que RASTI en sales amb efectes de reverberació, com són els locals públics. I gràcies a què els seus resultats són molt més acurats, en un futur podria acabar substituint al mètode de RASTI.

El mètode de STIPA, de la mateixa manera que el STI, utilitza les 7 bandes d'octava i 12 índexs de modulació són mesurats en total. En canvi, el temps de mesura és aproximadament el mateix que el RASTI (uns 10 o 15 segons).

### 3.4 Short-Time Objective Intelligibility (STOI)

A la figura 3.4 es pot veure il·lustrada l'estructura bàsica de STOI, on es pot observar que tenim com entrades el senyal de veu sense soroll de fons i el senyal de veu amb soroll de fons. La sortida de STOI és un valor que té una relació directa amb la intel·ligibilitat. Un rang de 10 KHz es utilitzat per tal de capturar la part més rellevant del senyal de veu.

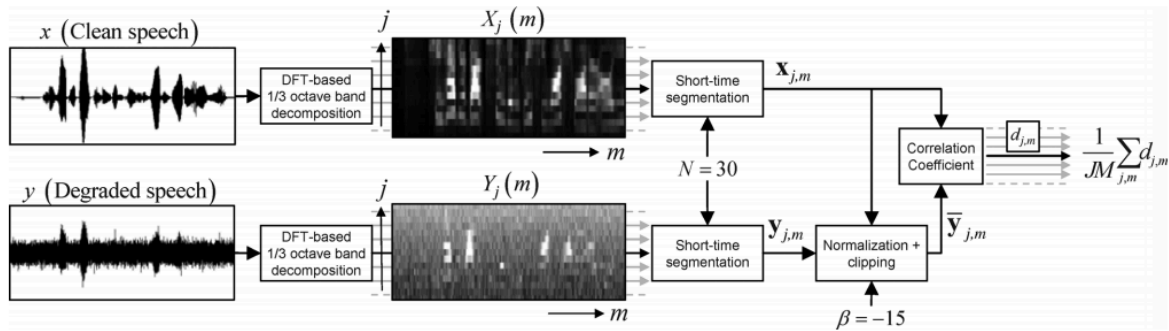


Figura 3.4 - Il·lustració de l'estructura bàsica de STOI

Primerament, els dos senyals són descompostos en temps i freqüència, per tal d'obtenir una representació simplificada. Això s'obté segmentant els dos senyals amb un *overlapping* del 50%, en l'aplicació de finestres de Hanning amb una llargada de 256 mostres. Abans de l'avaluació, les regions de silenci no tindran una contribució en el càlcul de la intel·ligibilitat, per tant, aquestes regions seran eliminades. Per aconseguir eliminar les zones de silenci, es trobarà la trama amb màxima energia de senyal de veu sense soroll, per tant, on l'energia és més petita, seran les zones on no trobem veu. Per tal de procedir a l'eliminació de les zones de silenci, les dues senyals seran reestructurades, eliminant totes les trames on l'energia del senyal de veu sense soroll de fons sigui 40 dB més petita que la trama de nivell màxim. Després de l'eliminació dels silencis, es filtren els senyals amb uns filtres de terç d'octava, en total s'utilitzen 15 bandes de terç d'octava, on la freqüència central més baixa és de 150 Hz i la freqüència central més alta és de 4300 Hz.

STOI compara el senyal amb soroll de fons amb un coeficient de correlació. Aquestes regions temporals son de 384 ms. Abans de calcular el coeficient de correlació, el senyal degradat amb soroll de fons se l'hi aplica una normalització per un possible *clipping*. Una vegada ja tenim feta aquesta normalització, es

calcula el coeficient de correlació per totes les bandes en totes les trames. El resultat final ve representat per la següent equació:

$$d = \frac{1}{J \cdot M} \sum_{j,m} d_{j,m}$$

M representa el número total de trames i J representa el número de bandes de terç d'octava.

## 5. Implementació

### 5.1 Tasques preliminars a la implementació

Abans de començar a entrar en matèria de la implementació del programa, s'ha de tenir en compte uns quants factors per comprendre com funcionaran els diferents càlculs necessaris dins del programa. Tot seguit es presentarà l'itinerari seguit per explicar com assolir els objectius del projecte.

Durant la introducció ja s'ha vist tots els fonaments bàsics d'acústica que s'utilitzaran durant tota la implementació, sabem com funciona l'orella humana, els fonaments bàsics de l'espectre del so, ja que s'utilitzen per crear filtres de terç d'octava. Durant la introducció també es va veure un resum ràpid de la sonoritat i les característiques principals de la veu humana. Un altre part a tenir en compte en la intel·ligibilitat és l'acústica arquitectònica, tot i que durant la implementació d'aquest problema no s'ha tingut en compte aquesta part, ja que no s'ha tingut en compte una sala en especial, sinó que ho s'ha fet d'una forma més genèrica.

Al següent tema s'ha definit la intel·ligibilitat i a més a més s'ha parlat de diferents mètodes per poder mesurar la intel·ligibilitat d'una forma objectiva, tot i que la intel·ligibilitat és un aspecte molt subjectiu. En aquest segon capítol és on s'ha parlat dels diferents índexs que hi han implementats al projecte, excepte del STI i els seus germans petits, RASTI i STIPA, ja que aquests índexs tenen en compte aspectes de l'acústica arquitectònica i durant aquest projecte no s'ha tingut en compte aquesta part.

L'objectiu del projecte és aconseguir una millora de la intel·ligibilitat de la paraula i quantificar objectivament amb els diferents índexs aquesta millora, a més a més de poder quantificar la millora de forma matemàtica, també s'ha incorporat un reproductor per tal de poder quantificar aquesta millora de forma subjectiva.

Un problema que se'ns planteja és que el soroll de fons és inevitable, i per tant, no es pot fer una millora que atenuï el soroll de fons, així que el que s'ha de fer

és aplicar una millora a la senyal de veu per tal d'aconseguir que la intel·ligibilitat millori.

## 5.2 Programari utilitzat

Per tal de crear el programa es podien utilitzar diferents programes i llenguatges de programació, però en el nostre cas s'ha decidit utilitzar el programa **Matlab** en versió R2012a amb el paquet d'eines **DSP System Toolbox**.

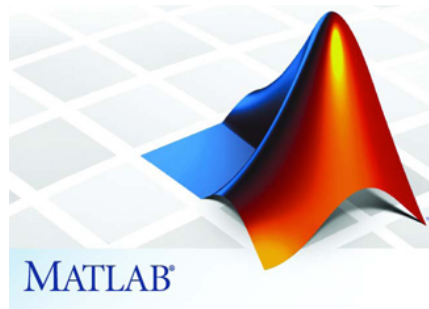


Figura 5.2 - Logotip del programa Matlab

S'ha utilitzat aquest programa gràcies a la familiarització que tenim amb el programa, per la pràctica que hem adquirit en les diferents assignatures que utilitzem aquest programa. A més a més, gràcies al paquet d'eines de DSP System Toolbox, tenim un gran ventall de funcions que són molt útils per a la realització d'aquest projecte. A la figura 5.2.1 podem observar una captura de l'eina per dissenyar filtres incorporada al paquet DSP System Toolbox.

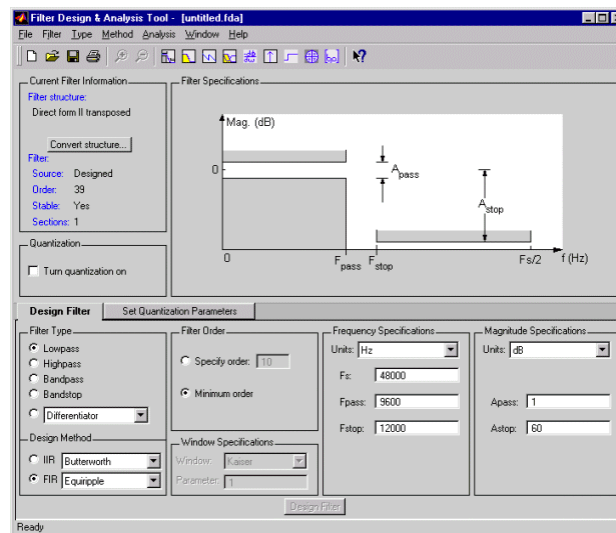


Figura 5.2.1 - Exemple de l'eina per dissenyar filtres amb DSP System Toolbox



## 5.3 Disseny del programa

Durant el següent capítol s'explica com s'ha fet el programa. S'exposaran les diferents eines que s'han utilitzat durant la implementació i com han estat implementades. Hem de recordar que s'han utilitzat funcions que s'inclouen dins de la llibreria *DSP System Toolbox*, per tant, per poder compilar el programa es necessari tenir aquest paquet instal·lat.

En aquesta part només presentarem d'una forma ràpida el funcionament i es definirà què fan i com les diferents funcions, però no es comentaran totes les línies del codi, per aquests a l'Annex A és troba tot el codi comentat.

### 5.3.1 Funcions dels índexs d'intel·ligibilitat

En aquesta part es definirà un per un els tres índexs que s'han implementat (AI, SII, STOI), s'explicarà com funcionen i com arriben al càlcul final.

```
· function A = AI(x, n, sn, pot)
```

Aquesta funció conté el codi pel càlcul de l'*Articulation Index*. Com a paràmetres d'entrada rep el senyal de veu, el senyal del soroll de fons, la relació senyal/soroll en dB i la potència de la senyal de veu en dB. Dins de la funció tenim un filtre que es divideix en 20 bandes, d'aquestes bandes s'ha parlat durant la definició de l'*Articulation Index* anteriorment. Amb aquests filtres, es divideixen les senyals de veu i soroll i es calcula la potència en dB que té cadascuna de les diferents bandes. Aquestes potències que té cada banda s'envien a la funció *artind*.

```
· function [AI,W,f]=artind(V, R, Be, type)
```

Aquesta funció es troba dins de la funció anterior. Aquí es reben 2 vectors amb les potències de cadascuna de les 20 bandes. V conté el vector amb els valors de la senyal de veu i Be conté el vector amb els valors de la senyal de soroll de fons. En aquest programa s'utilitza el tipus *d'Articulation Index* de Kryter.

```
· function Sn = SII_1(x,n,sn,pot)
```

Aquesta funció conté el codi pel càlcul del SII. Com en el cas anterior, es

tracta d'una preparació per tal de fer el càlcul de l'índex. Com a paràmetres d'entrada rep el senyal de veu, el senyal del soroll de fons, la relació senyal/soroll en dB i la potència de la senyal de veu en dB. Dins de la funció trobem un filtre de 18 bandes de terç d'octava, des de la freqüència central de 160 Hz, fins a 8000 Hz. Amb aquest filtre es fa el càlcul de la potència en dB de cadascuna de les bandes, tant pel senyal de veu com pel senyal de soroll. Aquestes potències de cada banda s'enviaran a la *funció SII*.

```
· function S = SII(varargin)
```

Com a paràmetres d'entrada s'envien a la funció les potències en dB del senyal de veu i del senyal de soroll de fons. Els altres valors queden tots per defecte. Per fer el càlcul del valor del SII, el programa segueix els passos descrits anteriorment en la descripció del *Speech Intelligibility Index*.

```
· function d = STOI_1(s, n, sn, Fs)
```

Aquesta funció té el codi pel càlcul del STOI. Com a paràmetres d'entrada rep la senyal de veu, la senyal de soroll de fons i la relació senyal/soroll, ja que la freqüència de mostreig és de 44100 Hz per defecte. Dins d'aquesta funció s'agafa el senyal i depenent del valor del SNR, s'aplica un filtre corrector al senyal de veu o a la senyal de soroll. Aquestes senyals s'envien a la funció *stoi*.

```
· function d = stoi(x, y, fs_signal)
```

Amb aquesta funció es fa el càlcul del valor del STOI. Com a paràmetres d'entrada rep el senyal de veu (x) i el senyal de veu sumada de la senyal de soroll de fons (y), a més a més de la freqüència de mostreig. Amb aquest paràmetres seguint els passos explicats a l'apartat 3.4, fa el càlcul del STOI.

### 5.3.2 Funcions dels filtres

En aquesta part es detallarà el funcionament de les funcions dels diferents filtres que s'utilitzen en aquest programa.

```
· function [hd, Nfc] = filtroAI()
```

Aquesta funció crea un filtre amb les bandes indicades a la *taula 3.1*. A la *figura 5.3.2.1* es pot observar el comportament del filtre, on apareixen les 20 diferents bandes diferenciades.

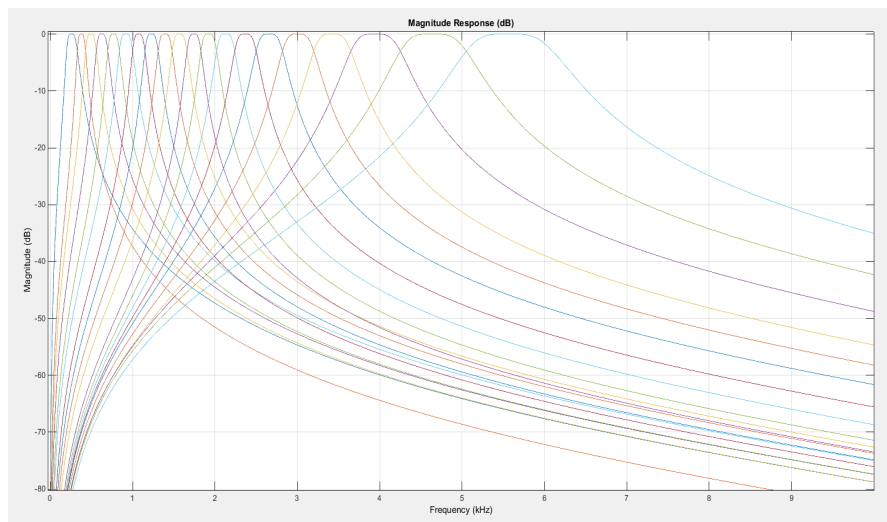


Figura 5.3.2.1 – Resposta en freqüència del filtre de l'Articulation Index

```
· function [Hd3, Nfc, F0] = filtro13octava()
```

Aquesta funció crea un filtre de terç d'octava. A la *figura 5.3.2.2* es pot observar el comportament del filtre de terç d'octava.

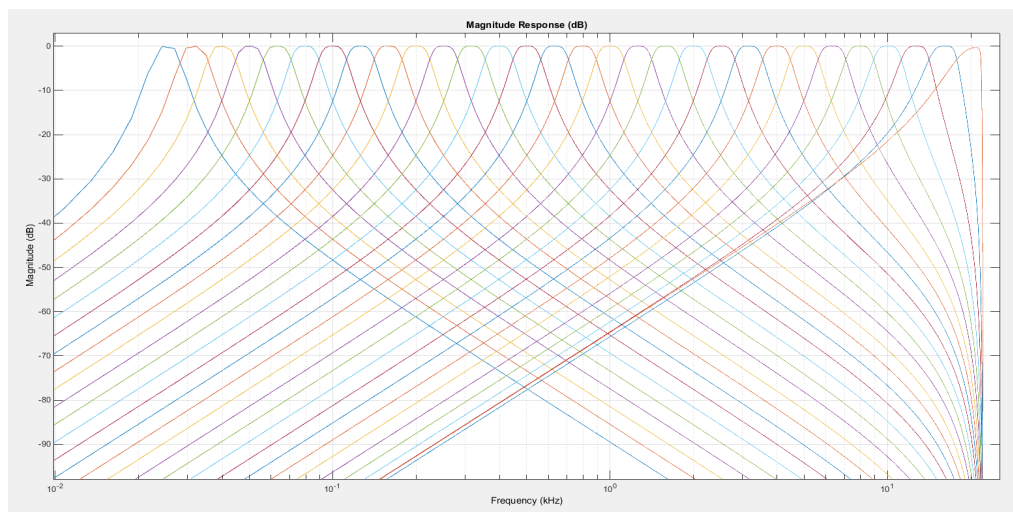
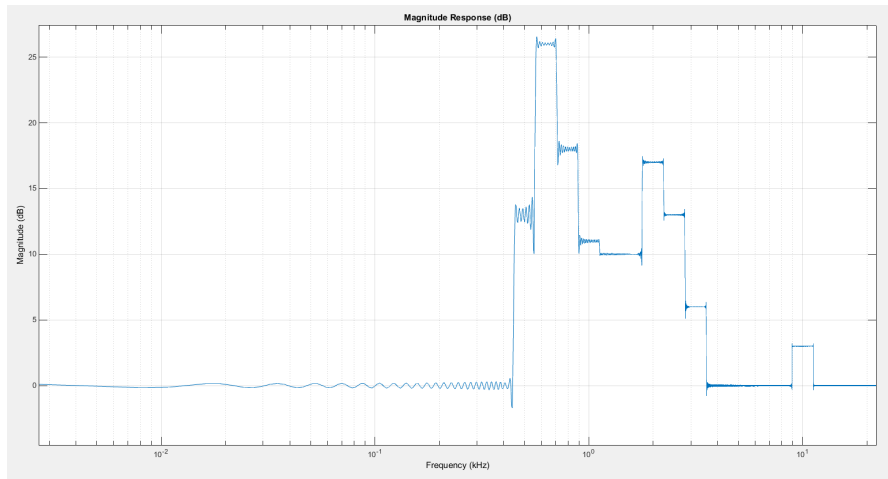


Figura 5.3.2.2 – Resposta en freqüència del filtre de terç d'octava

```
· function filtre = crea_filtro (ponderacion, orden,
fs, liminf, limsup)
```

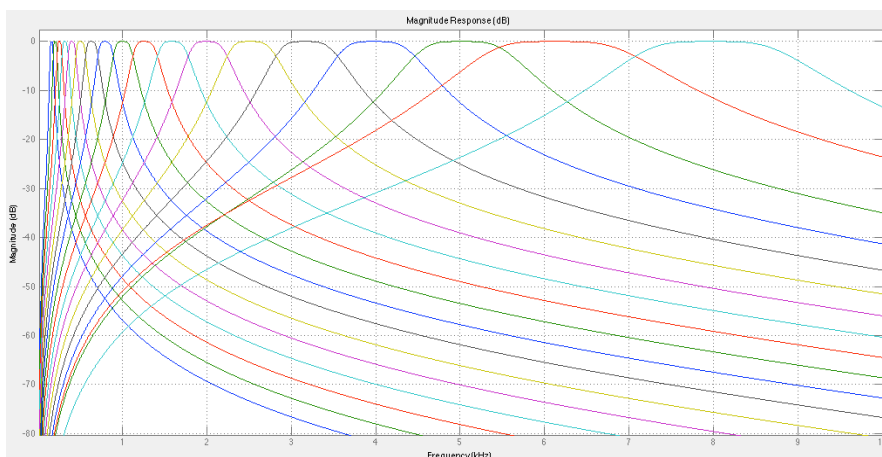
Aquesta funció crea el filtre per la millora de la senyal de veu. Aquesta funció rep com a paràmetres d'entrada un vector amb les ponderacions, l'orde del filtre, la freqüència de mostreig i els límits inferior i superior de freqüència. Amb aquests paràmetres crea un filtre de terç d'octava amb valors iguals al vector de les ponderacions.



**Figura 5.3.2.4 – Exemple de la resposta en freqüència d'un filtre que s'aplica a una millora**

```
· function [hd Nfc] = filtroSII()
```

Aquesta funció crea un filtre de terç d'octava amb les 18 bandes que comencen a la freqüència central de 160 Hz i acaba en la freqüència central de 8000 Hz. A la figura 5.3.2.3 podem veure el comportament d'aquest filtre.



**Figura 5.3.2.4 - Resposta en freqüència del filtre del SII**

### 5.3.3 Codi principal

En aquest apartat es veurà el codi més important escrit per a la creació del programa. Si es vol veure tot el codi escrit, es pot consultar a l'Annex A.

```
· function tag_element_CreateFcn(hObject, eventdata,  
handles, varargin)
```

Aquesta funció es crea per a tots els textos, tant editables com estàtics, i menús. Es crea automàticament amb la creació de l'objecte.

```
· function guide1_OpeningFcn(hObject, eventdata,  
handles, varargin)
```

Els handles són compartits per totes les funcions, per tant, en aquesta part és on s'inicialitzen totes les variables que es faran servir dins de la interfície que s'ha creat. Totes les variables que s'han creat estan buides, per tant, s'aniran omplint mentre anem cridant les diferents funcions. Aquestes variables poden utilitzar-se en les diferents funcions dintre de la interfície, gràcies a aquesta funcionalitat, les variables que s'inicialitzen a aquesta part són totes les que tenen alguna cosa a veure amb els diferents senyals, de manera que es puguin llegir o crear en alguna funció i després utilitzar-la o reproduir-la en alguna altra funció.

```
· function varargout = guide1_OutputFcn(hObject,  
eventdata, handles)
```

Aquesta funció serveix per retornar a la línia de comandes les dades necessàries per utilitzar-les fora de la interfície. En el nostre cas aquesta funció estarà buida, ja que totes les dades s'utilitzen dintre de la mateixa interfície.

```
· function figure1_CloseRequestFcn (hObject,  
eventdata, handles)
```

Aquesta funció s'executa quan l'usuari tanca el programa. La utilitat d'aquesta funció és de treure un missatge d'avís per saber si s'està segur de voler tancar el programa.

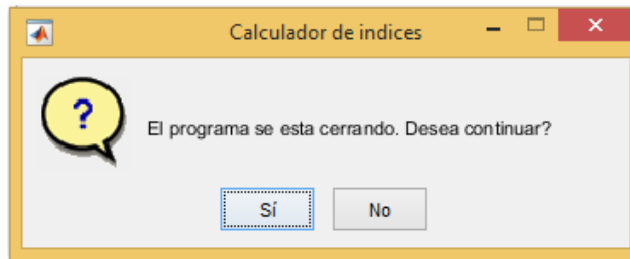


Figura 5.3.3.1 – Finestra per la confirmació de tancar el programa

```
· function Abrir_Ruido_Callback (hObject, eventdata, handles)
```

Aquesta funció serveix per carregar l'arxiu del senyal de soroll de fons que es desitja per fer el càlcul. Una vegada es prem el botó per tal de carregar el senyal de soroll, s'obre un navegador pel qual es pot navegar i escollir el soroll que es vulgui. Una vegada el senyal es carrega a una variable ja creada anteriorment, fa que el text que indica que el soroll ha estat carregat aparegui com visible.

```
· function Abrir_Senyal_Callback (hObject, eventdata, handles)
```

Aquesta funció fa exactament el mateix que la funció anterior, però en comptes de carregar el senyal de soroll de fons, llegeix el senyal de veu.

```
· function SNRdB_Callback (hObject, eventdata, handles)
```

Aquesta funció té dues utilitats. Primerament serveix per posar el valor del SNR en dB, i com a segona funció, serveix per què en el cas que el valor s'escriui de forma lineal, en l'altre *textbox*, transformi el valor lineal a dBs.

```
· function SNR_Callback (hObject, eventdata, handles)
```

Aquesta funció fa exactament el mateix que l'anterior per a l'inrevés, l'hi pots escriure el valor del SNR en lineal o transforma el valor en dB a lineal.

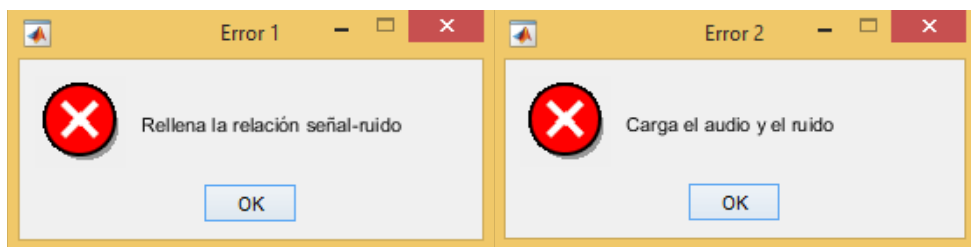
```
· function play1_Callback (hObject, eventdata, handles)
```

Aquesta funció reproduïx el senyal d'àudio sense haver aplicat la millora. Incorpora un missatge d'error en el cas de que no s'hagi calculat aquest senyal.

Les altres diferents funcions que tenen el nom de play i algun número, fan una reproducció dels altres senyals millorats.

```
· function calcular_Callback (hObject, eventdata, handles)
```

Aquesta funció es tracta d'un botó que serveix per fer el càlcul de l'índex desitjat abans d'aplicar cap millora. El primer que fa aquesta funció és llegir el valor del SNR i la potència de la senyal. Una vegada ha fet això, comprova si s'han carregat la senyal de veu i la senyal de soroll de fons. En el cas que alguna d'aquestes senyals o el SNR estigui buit, apareixeran els possibles errors representats a la *figura 5.3.3.2*

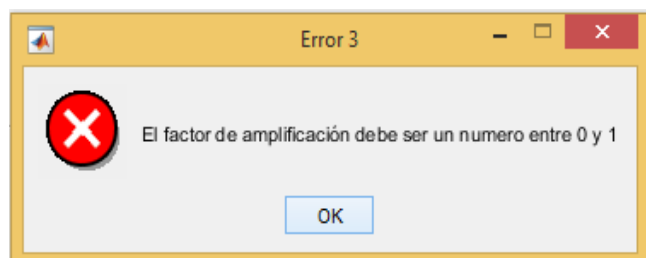


**Figura 5.3.3.2 – Possibles missatges d'error**

Una vegada tenim tot carregat de forma correcta, s'aplica una normalització tant al senyal de veu com al senyal de soroll per tal d'igualar el nivell dels dos senyals. Després d'aplicar aquesta igualació de potència dels senyals, s'aplica una correcció de senyal que ve indicada pel valor indicat al SNR. Quan es tenen els senyals preparats de forma correcta, es procedeix a fer el càlcul de l'índex de intel·ligibilitat, que depenen de l'índex que s'hagin seleccionat anteriorment, es farà un càlcul o altre. Finalment es sumen els dos senyals i es guarden en una variable per poder reproduir-la posteriorment.

```
· function calculamejora_Callback (hObject, eventdata, handles)
```

Aquesta funció conté la implementació per realitzar la millora del senyal de veu i quantificar la millora amb el càlcul de l'índex d'intel·ligibilitat de la paraula del senyal amb la veu millorada. Per fer el càlcul, el primer que es tindrà en compte serà quin tipus de millora es vol aplicar. Una vegada es té el tipus de millora, s'ha de tenir en compte altres variables com són el factor d'amplificació (en el cas d'aplicar la millora 3) o la diferència mínima (en el cas d'aplicar la millora 1). Un altre factor a tenir en compte és el rang de freqüències on es volen aplicar la millora, indicant la freqüència mínima i màxima.



**Figura 5.3.3.3 – Missatge d'error que apareix si el valor del factor d'amplificació no es correcte**

Una vegada es tenen tots els paràmetres, s'aplica la millora seleccionada, es fa el càlcul de la intel·ligibilitat i s'indica el valor a la finestra correcta, a més a més d'indicar el valor de la amplificació total que s'ha aplicat al fer la millora. Finalment com en el cas anterior, la senyal sense millora i la senyal millorada es guarden cadascuna a una variable diferent per tal de poder reproduir-les posteriorment.

```
· function compara_Callback (hObject, eventdata, handles)
```

Aquesta funció conté la implementació per realitzar una comparació entre les diferents millores. Amb aquesta funció es pot comparar la millora 2 respecte les millores 1 o 3, depenent de quina de les dues millores s'indiqui. Una vegada seleccionada amb quina de les dues millores es vol comparar, s'aplica la millora triada. Una vegada s'ha aplicat aquesta millora, es calcula l'amplificació total que ha aplicat aquesta millora, i amb això s'aplica la millora 2 amb una amplificació total igual a l'anterior. Després de tenir els dos senyals millorats en dues variables diferents, es realitza el càlcul de l'índex d'intel·ligibilitat triat i



s'indiquen els valors calculats a les finestres adequades. Finalment com en els casos anteriors, la senyal sense millora i els dos senyals millorats es guarden cadascun a una variable diferent per tal de poder reproduir-les posteriorment.

### 5.3.4 Funcions secundaries

En aquesta part es detallen tota la resta de funcions que s'utilitzen per dur a terme el funcionament del programa.

```
· function dbdif = calculo_ponderacion (dbpowsenyal,  
dbpowruido, tipo, difmin, factamp, amp, maxi)
```

Aquesta funció retorna un vector amb els dB d'ampliació necessària per cadascuna de les 30 bandes de terç d'octava. Aquest càlcul es realitza depenent de la potència de cada banda del senyal de veu, de la potència del senyal de soroll, del tipus de millora aplicada, de la diferència mínima el factor d'amplificació i el tipus de millora que es desitja per aplicar la millora 2. La diferència mínima només es té en compte si s'aplica la millora 1, mentre que el factor d'amplificació és necessari en el cas d'aplicar la millora número 2 i 3, cada millora té el seu factor d'amplificació.

```
· function dbpow = calculo_potencia (x, volsenyal)
```

Amb aquesta funció es fa el càlcul de la potència de cada banda de terç d'octava tot indicant el senyal que volem calcular i la potència d'aquest senyal.

```
· function s_corregida = compara(s, potamp, liminf,  
limsup, deltaLp)
```

Aquesta funció retorna el senyal de veu corregit aplicant la millora 2 tenint en compte l'amplificació total que s'ha utilitzat a l'altre millora amb què comparar.

```
· function s_corregida = correccion(s, n, potamp, tipo,  
difmin, liminf, limsup, factamp, amp, maxi)
```

Aquesta funció realitza la millora del senyal de veu. Rebut els dos senyals i la potència d'amplificació fa el càlcul de potència per totes les bandes, amb això fa el càlcul de la ponderació amb la funció descrita anteriorment de *calculo\_ponderacion*. Finalment, una vegada es té calculat el vector amb els valors pels que s'ha d'amplificar cada banda, s'aplica un filtre amb aquests valors a la senyal original per tal d'aconseguir una senyal millorada.

```
· function [ s, n ] = corrección_volumen(s, n, sn)
```

Amb aquesta funció s'aplica una correcció del volum dependent del valor del SNR. D'una forma senzilla, aquesta funció només aplica un filtre corrector que atenua la senyal que té menys potència dependent del SNR, per tant, si el SNR és positiu, significarà que el senyal de veu és més alt que el de soroll, i per tant s'aplicarà una atenuació del valor del SNR al senyal de soroll de fons. En el cas que el SNR sigui negatiu es farà a l'inrevés.

```
· function [ s, n, potamp ] = igualación_potencia (x, y, pot)
```

Aquesta funció conté la implementació per fer el càlcul de la igualació de potència dels dos senyals. Rebut els dos senyals originals i la potència que es volen igualar ambdós senyals, treu el senyal de soroll i de senyal amb el mateix nivell de potència, a més a més del nivell de potència necessari per adquirir el nivell que l'hi hem indicat com entrada.

```
· function increment_find = Increment (DeltaLp, liminf, limsup, dbpowsenyal)
```

Aquesta funció retorna el vector amb els nivells que s'han d'amplificar les diferents bandes per tal d'aconseguir una amplificació total donada per la variable DeltaLp. Aquesta amplificació es realitza de forma plana dins dels límits de freqüència mínima i màxima. Fora d'aquests límits no hi haurà amplificació de cap tipus.

```
· function scor = normalizacion (s)
```

Aquesta funció realitza una normalització del senyal que s'aplicarà just abans de la reproducció del senyal.

## 5.4 Millores proposades

En aquest apartat s'explicaran les diferents millores que s'han implementat durant aquest projecte. Bàsicament s'han implementat tres millores diferents on cadascuna d'elles fa una amplificació per bandes de freqüència d'una forma diferent. Per tal d'entendre millor les gràfiques, a la taula 5.4 es poden veure les freqüències centrals que corresponen a cada número de banda

Número de banda	Freqüència central (Hz)	Número de banda	Freqüència central (Hz)
1	25	16	794
2	32	17	1000
3	40	18	1259
4	50	19	1585
5	63	20	1995
6	79	21	2512
7	100	22	3162
8	126	23	3981
9	158	24	5011
10	200	25	6310
11	251	26	7943
12	316	27	10000
13	398	28	12589
14	501	29	15850
15	631	30	19953

Taula 5.4 – Correspondencia del número de banda amb la freqüència central

### 5.4.1 Primera millora

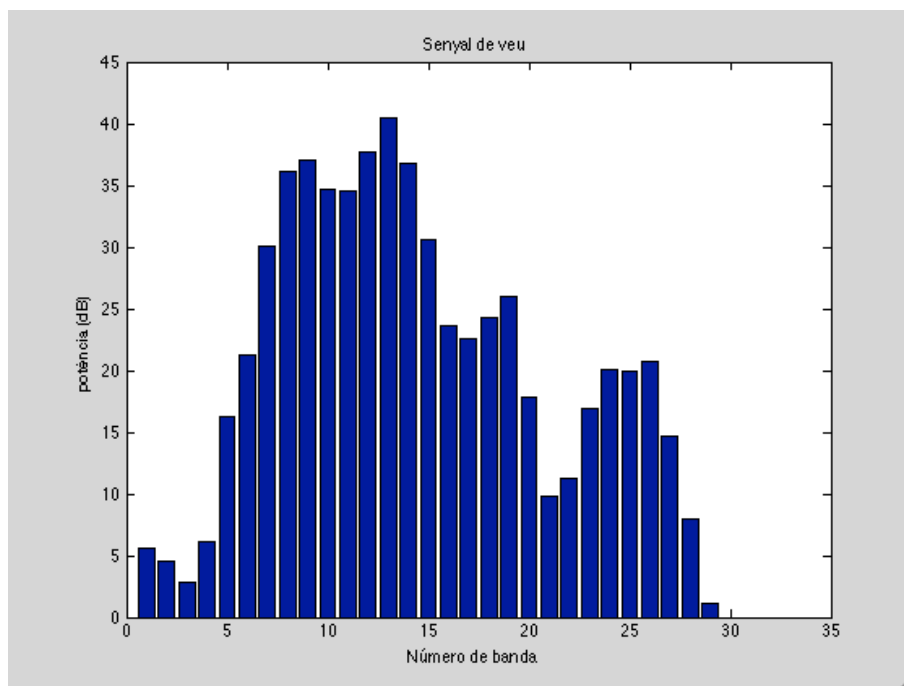
La primera millora és la que es selecciona al programa com la millora número 1. Aquesta millora amplifica el senyal de veu a les bandes on el senyal de soroll té més potència que el senyal de veu, quan no és així i la senyal de veu té més

potència que el senyal de soroll, la senyal de veu no es veurà amplificada. Això és així quan la diferència mínima es troba a 0 dB, si la diferència mínima té un valor positiu, farà el mateix, però a més a més amplificarà totes les bandes que es troben dins del rang de freqüències per aquest valor.

A continuació veurem un exemple de com funciona aquesta millora amb els següents paràmetres:

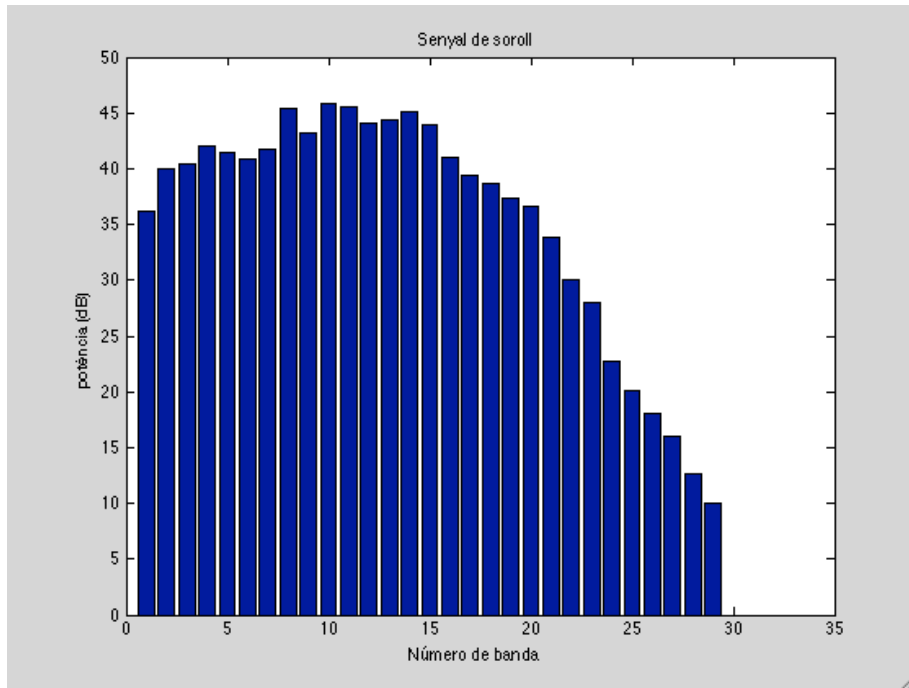
- SNR = -10 dB
- Rang de freqüències: de 500 a 12000 Hz
- Diferència mínima: 0

Primerament fa un anàlisi de la senyal de veu. Això ho podem observar a la *figura 5.4.1.1*.



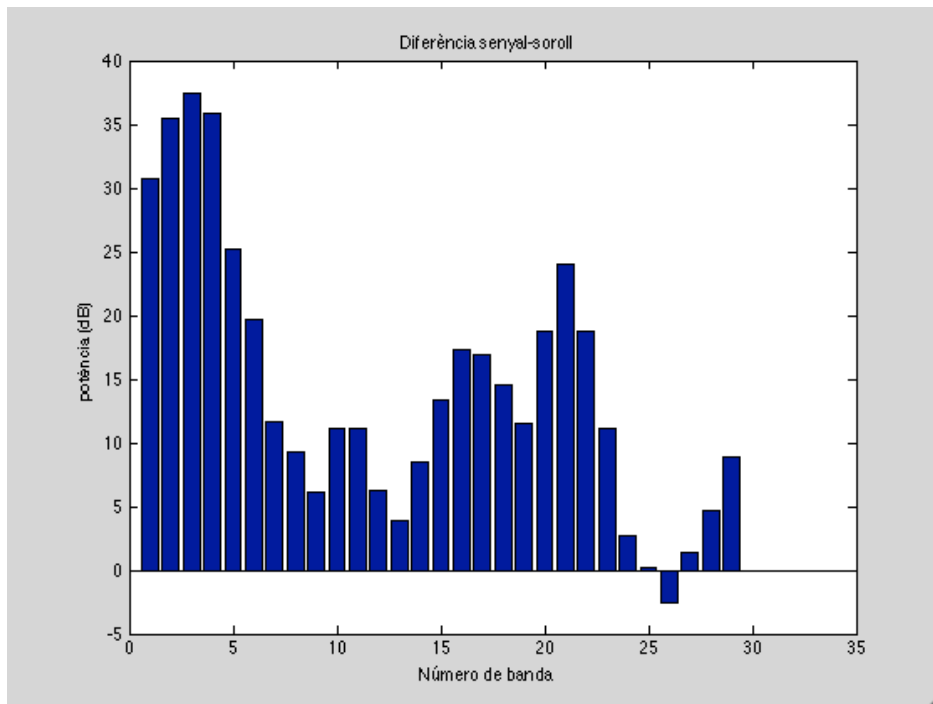
**Figura 5.4.1.1 – Anàlisi espectral del senyal de veu**

Una vegada s'ha fet l'anàlisi per cada banda de terç d'octava de la senyal de veu, s'ha de fer el mateix pel senyal de soroll de fons. Es pot veure el resultat d'aquest anàlisi a la *figura 5.4.1.2*.



**Figura 5.4.1.2 – Anàlisi espectral del senyal de soroll de fons**

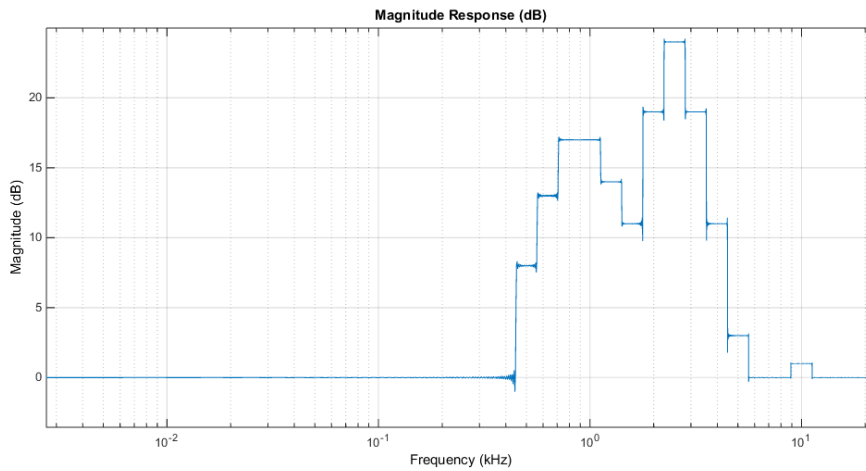
Una vegada es tenen els espectres dels dos senyals, s'agafen els dos senyals i es resta la potència del senyal de veu a la potència de la senyal de soroll. El resultat de la diferència per aquest exemple es pot observar a la figura 5.4.1.3.



**Figura 5.4.1.3 – Diferències de nivell entre les diferents bandes**

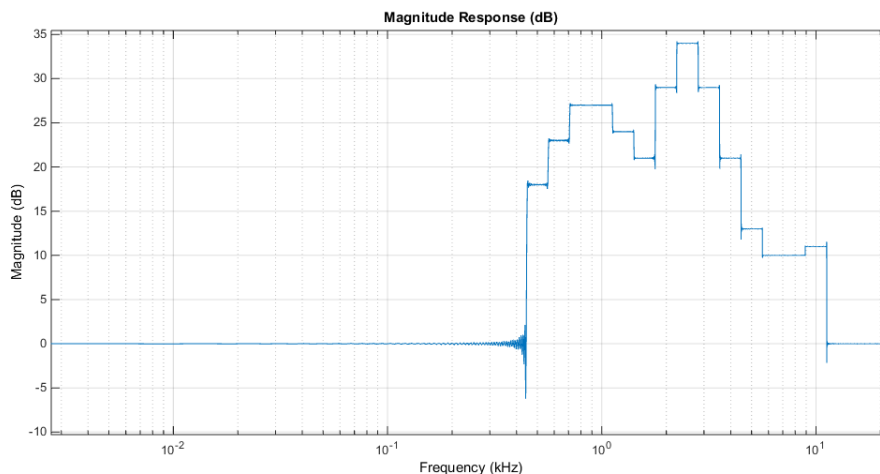
Una vegada es tenen les diferències de potència per cadascuna de les bandes, aquesta millora el que fa és que per totes les bandes que estan dins del rang

de freqüències on volem actuar, aplicarà una amplificació del valor d'aquesta banda, quan el valor sigui negatiu, no aplicarà cap amplificació. Totes les bandes que estiguin fora del rang no tindran cap canvi. Es pot veure un exemple del filtre que s'utilitza pel nostre exemple a la *figura 5.4.1.4*.



**Figura 5.4.1.4 – Filtre de millora amb diferència mínima de 0 dB**

Com es pot veure a la figura anterior, a les bandes menors de 500 Hz i les majors de 12 KHz, no les amplifica, i a les altres s'aplica una amplificació del valor de la diferència per aquesta banda. En el cas de triar una diferència mínima que sigui major a 0, a les bandes que es troben dins del rang, se'ls hi aplicarà una amplificació plana amb una potència d'aquest valor. Com a exemple, es pot observar a la figura 5.4.1.5 l'aplicació de la millora anterior amb l'única diferència que s'ha introduït un valor de 10 dB a la variable de diferència mínima.



**Figura 5.4.1.5 – Filtre de millora amb diferència mínima de 10 dB**

Com es pot veure, el filtre és exactament igual amb la diferència que dins del rang s'aplica una amplificació addicional de 10 dB per cadascuna de les bandes.

### 5.4.1 Segona millora

Aquesta millora és la més senzilla de totes, l'únic que fa és aplicar una amplificació plana dins del rang de freqüències. Per realitzar aquesta millora hi ha dues opcions, la primera és indicar que es desitja una amplificació màxima, d'aquesta manera, el valor pel què amplifica és el valor màxim de la diferència, per tant, en el cas anterior aplicaria un filtre amb una amplificació de 38 dB per totes les bandes. A la *figura 5.4.1.6* es pot observar com quedaria el filtre aplicant aquesta millora amb l'amplificació màxima.

L'altre opció per aquesta millora es indicar el valor per el que es desitja amplificar cadascuna de les bandes. D'aquesta manera, en el cas d'indicar un factor d'amplificació de 5 dB, cadascuna de les bandes que es trobin dins del rang freqüencial on s'aplica el filtre, tindran una amplificació de 5 dB.

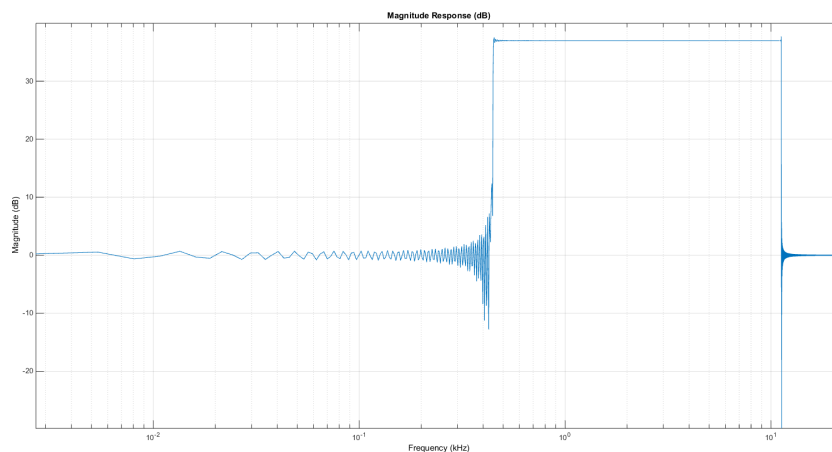


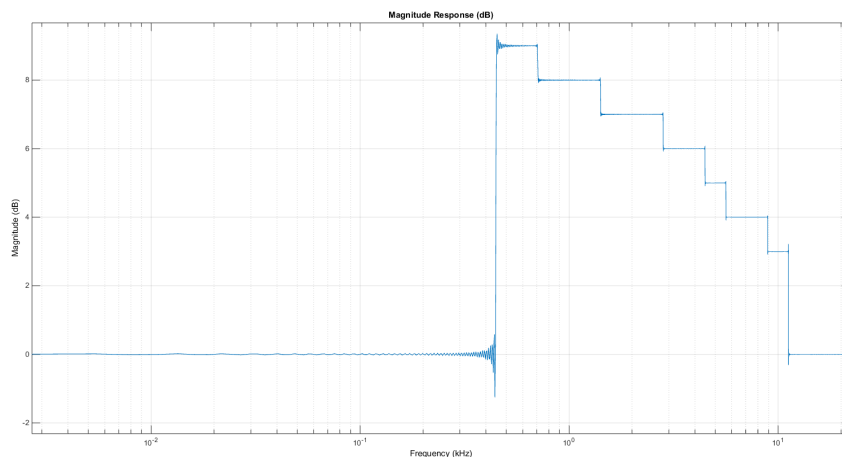
Figura 5.4.1.6 – Exemple de filtre per la segona millora

### 5.4.1 Tercera millora

Aquesta millora només té en compte el soroll de fons per tal d'aplicar la millora. Primer de tot fa un anàlisi del senyal de soroll de fons. Una vegada ha fet aquest anàlisi, multiplica els valors de cadascuna de les diferents bandes pel valor del factor d'amplificació, aquest valor ha d'estar entre 0 i 1, sinó el

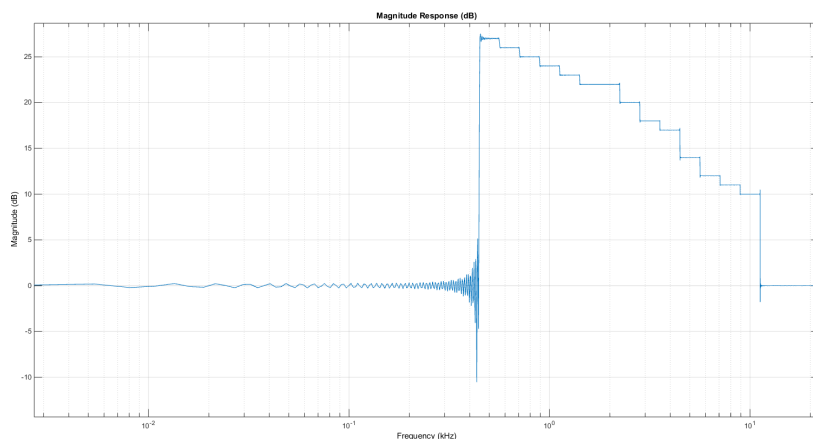
programa donarà un error. La forma de l'espectre que s'envia per crear el filtre és exactament igual que el de l'espectre del soroll de fons, amb l'única diferència que els valors són els mateixos multiplicats pel factor d'amplificació

Com en les millores anteriors, aquest filtre només s'aplica al rang de freqüències seleccionat. Fora d'aquest rang, la senyal es queda igual. A la *figura 5.4.1.2* es pot observar com seria el filtre que s'aplica a la senyal de veu utilitzant l'exemple anterior i amb un factor d'amplificació de 0,2.



**Figura 5.4.1.2 – Filtre de la segona millora amb factor d'amplificació de 0,2**

A la *figura 5.4.1.3* es pot observar que en augmentar el factor d'amplificació, també augmenta l'amplificació per cadascuna de les bandes, en aquest cas s'aplica un factor d'amplificació de 0,6.



**Figura 5.4.1.3 – Filtre de la segona millora amb factor d'amplificació de 0,6**



## 5.5 Funcionament del programa

En aquest apartat s'explicarà d'una manera més gràfica el funcionament final del programa, i es veuran totes les opcions que es poden utilitzar.

A la figura 5.3 es pot veure una il·lustració de com és el programa principal, a partir d'aquesta imatge s'anirà explicant el funcionament bàsic del programa i per que serveixen els diferents paràmetres que es poden canviar.

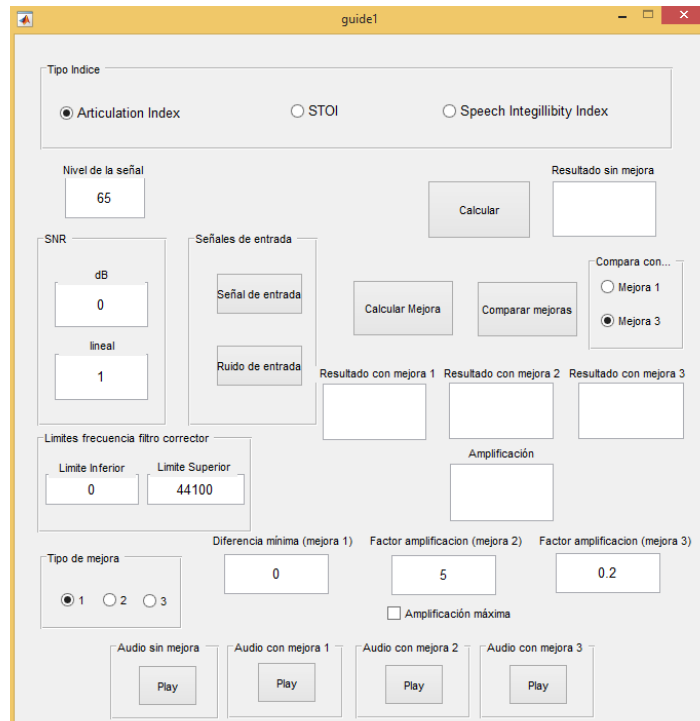


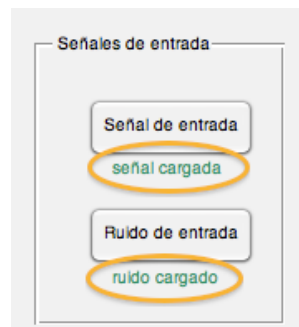
Figura 5.5 - Il·lustració del programa principal

Primer de tot, a la part de dalt, es pot triar entre els 3 diferents índexs que s'han implementat en aquest programa. Les opcions possibles són *Articulation Index (AI)*, *STOI* i *Speech Intelligibility Index (SII)*.

Una vegada s'ha triat el mètode que es vol calcular, es pot variar el nivell de la senyal, això és important per indicar el nivell dels dos senyals, si per exemple es posa un nivell molt petit, la intel·ligibilitat seria molt baixa degut al llimitar d'audició, en canvi, si el nivell és molt alt, podríem tenir una distorsió que també faria que la intel·ligibilitat de la paraula baixés. Per tant, per aquesta variable el més recomanable és posar un valor entre 60 i 90 dB, en aquest projecte per l'estudi de les millores s'utilitza el valor de 75 dB.

El següent paràmetre que s'ha de decidir és el SNR, per aquest paràmetre tenim dues opcions, la primera de totes i més fàcil és posar el valor en dB, però és interessant que també és possible indicar aquest valor de forma lineal. Cada vegada que es canvia un dels dos valors, quan es clica en qualsevol altre lloc, s'actualitza automàticament l'altre valor. El valor del SNR serveix per indicar la relació de nivells entre el senyal de veu i el senyal de soroll, de tal manera que si el valor del SNR és positiu, el senyal de veu tindrà més potència que el senyal de soroll i si el valor del SNR és negatiu, serà exactament a l'inrevés.

Després d'això s'han de carregar les diferents senyals, per fer això s'han implementat dos botons, un que serveix per carregar el senyal de veu i l'altre serveix per carregar el senyal de soroll de fons. Cada vegada que es pren un botó, apareix un navegador on es pot triar el senyal a carregar, com a puntualització, dir que els senyals han de ser de tipus .wav i en format mono. Una vegada s'ha carregat el senyal, apareix un text de color verd que indica que el senyal ha estat carregat correctament, a la *figura 5.3.1* es pot veure un exemple gràfic.



**Figura 5.5.1 - Il·lustració del text que indica que el senyal ha estat carregat correctament**

Amb això ja es tenen els diferents paràmetres per fer el càlcul de la intel·ligibilitat de la senyal sense millorar. Per fer això només s'ha de prémer el botó de *Calcular* i ja sortirà el valor a la finestra de *Resultado sin mejora*.

Després de veure els paràmetres que s'utilitzen per fer el càlcul de la senyal abans d'aplicar la millora, ja es pot parlar dels paràmetres que es poden ajustar per aplicar una millora a la senyal de veu. El primer que s'ha de tenir en compte és el rang de freqüències on s'aplicarà el filtre corrector. Per dur a terme això, es pot introduir el límit inferior i superior, de tal manera que el filtre que

s'aplicarà per fer la millora només actuarà en aquest rang, i la resta deixarà el senyal de la mateixa forma que al senyal original.

En un altre panell, es troba l'opció que ens permet triar el tipus de millora que es vol aplicar. Si es tria la millora número 2, s'haurà d'indicar el valor per el que volem amplificar cada banda o seleccionar si volem aplicar l'amplificació de la màxima diferència. En el cas de triar la millora número 1, serà necessari indicar la diferència mínima que es vol incrementar la senyal millorada respecte del soroll. I si es vol es aplicar la millora número 3, s'haurà d'escriure el valor que es desitgi del factor d'amplificació, un valor que haurà d'estar entre 0 i 1.

Una vegada s'han escrit les variables, només s'haurà de pressionar el botó de *Calcular mejora*. D'aquesta manera, sortirà el valor en la finestra adequada, ja que cada millora té una finestra diferent per escriure el resultat del càlcul. A més a més del resultat de l'índex d'intel·ligibilitat, sortirà en un altre finestra el resultat de l'amplificació total que ha aplicat el filtre corrector, així es podrà tenir una idea de si podria aparèixer una distorsió.

Un altre opció que té el programa és la de poder realitzar una comparació entre les millores 1 i 3 amb la millora 2. Per fer això, només s'ha de triar en el panel *Compara con...* si es vol fer la comparació amb la millora 1 o amb la 3. Una vegada s'ha fet l'elecció, s'ha de pressionar el botó *Comparar mejoras* i sortiran els valors de la intel·ligibilitat i de l'amplificació total que han aplicat les dues millores a les finestres adequades.

Per finalitzar, a la part baixa podem trobar 4 reproductors de so, on cadascun dels reproductors serveix per poder escoltar el senyal original i les diferents millores que s'han aplicat, i d'aquesta forma poder fer una comparació subjectiva dels valors que han donat els índex d'intel·ligibilitat.

## 6. Resultats

En aquest capítol s'exposen els resultats que donen els diferents índexs a l'hora d'aplicar les millores. Durant aquest apartat s'expondran els resultats primerament fent una comparació dels diferents índexs d'intel·ligibilitat i seguit d'aquesta part, es presentarà una discussió de la comparació entre les diferents millores que es poden aplicar.

Per entendre el que s'estarà comparant, primer de tot s'ha d'entendre quines seran les senyals amb els que es calcularan aquests resultats. El primer que s'ha de tenir en compte és el senyal de veu, per tots els càlculs s'utilitzarà el mateix senyal de veu, que tracta d'una simple gravació d'un senyal de veu humà. Seguidament, s'han de tenir en compte els diferents sorolls de fons que utilitzarem:

- Ambient Aeroport: és tracta d'una gravació del soroll de fons a la terminal d'un aeroport.
- Soroll blanc: és un tipus de soroll que té la característica de que totes les freqüències posseeixen la mateixa potència. La densitat espectral de potència es una constant, es a dir, la seva gràfica es totalment plana.

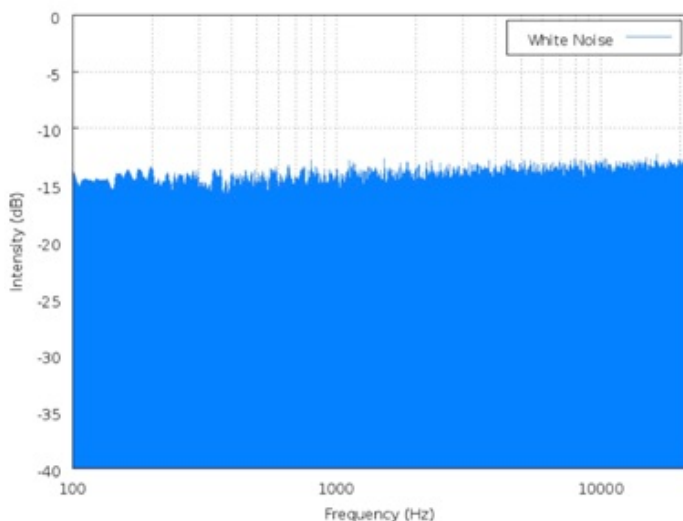
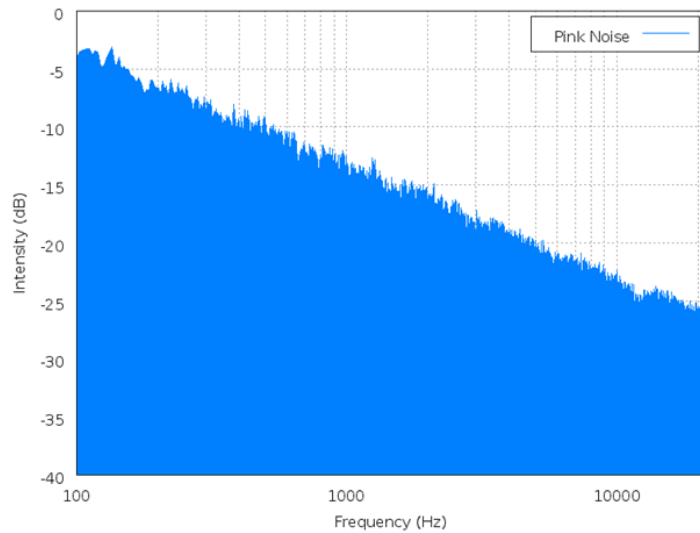


Figura 6.1 – Densitat espectral del soroll blanc

- Soroll rosa: és tracta d'un tipus de soroll que té la principal característica de que la densitat espectral es inversament proporcional a la freqüència. Quan aquest senyal es visualitza a un analitzador de filtres d'octava, es veu com totes les bandes tenen el mateix nivell sonor, cosa que es certa

ja que l'amplada de banda per les freqüències altes és més gran que a les freqüències petites



**Figura 6.2 – Densitat espectral del soroll rosa**

A la *taula 6.1* es poden observar els resultats que donen els diferents índexs per a les diferents millores implementades amb diferents sorolls de fons. En els resultats es pot veure el valor que dona cada índex i entre parèntesi l'amplificació total (en dB) que aplica cadascuna de les millores. El primer valor que trobarem serà el del càlcul del senyal original, després d'això vindran els resultats amb l'aplicació de les diferents millores indicant els valors que hem utilitzat per cada millora.

	Diferència mínima	Factor amplificació	Ambient aeroport			Soroll blanc			Soroll rosa		
			AI	STOI	SII	AI	STOI	SII	AI	STOI	SII
<b>Sense millora</b>	-	-	0,274	0,734	0,34	0,3	0,68	0,31	0,215	0,626	0,224
<b>Millora 1</b>	<b>0</b>	-	0,295 (0,3)	0,787 (0,3)	0,463 (0,3)	0,372 (1,4)	0,784 (1,4)	0,472 (1,4)	0,267 (0,9)	0,75 (0,9)	0,4 (0,9)
	<b>5</b>	-	0,449 (4,1)	0,884 (4,1)	0,598 (4,1)	0,478 (5,4)	0,9 (5,4)	0,575 (5,4)	0,393 (4,8)	0,865 (4,8)	0,516 (4,8)
	<b>10</b>	-	0,611 (8,6)	0,956 (8,6)	0,717 (8,6)	0,588 (10,1)	0,956 (10,1)	0,663 (10,1)	0,545 (9,3)	0,928 (9,3)	0,613 (9,3)
	<b>15</b>	-	0,775 (13,4)	0,971 (13,4)	0,81 (13,4)	0,7 (14,9)	0,979 (14,9)	0,734 (14,9)	0,68 (14,2)	0,956 (14,2)	0,691 (14,2)
	<b>20</b>	-	0,913 (18,4)	0,976 (18,4)	0,827 (18,4)	0,81 (19,9)	0,99 (19,9)	0,735 (19,9)	0,8 (19,2)	0,97 (19,2)	0,7 (19,2)
<b>Millora 2</b>	-	<b>Amplificació màxima</b>	0,999 (24,8)	0,98 (24,8)	0,8 (24,8)	0,97 (34,8)	0,998 (34,8)	0,675 (34,8)	0,91 (26,8)	0,973 (26,8)	0,667 (26,8)
<b>Millora 3</b>	-	<b>0,2</b>	0,548 (7)	0,9 (7)	0,545 (7)	0,49 (7,2)	0,856 (7,2)	0,499 (7,2)	0,491 (9)	0,841 (9)	0,457 (9)
	-	<b>0,4</b>	0,836 (15,7)	0,957 (15,7)	0,71 (15,7)	0,71 (16,3)	0,956 (16,3)	0,66 (16,3)	0,801 (20,8)	0,951 (20,8)	0,643 (20,8)
	-	<b>0,6</b>	0,997 (24,4)	0,975 (24,4)	0,792 (24,4)	0,878 (25,3)	0,993 (25,3)	0,699 (25,3)	0,965 (31,8)	0,982 (31,8)	0,658 (31,8)
	-	<b>0,8</b>	1 (32,8)	0,986 (32,8)	0,793 (32,8)	0,975 (34,7)	0,999 (34,7)	0,654 (34,7)	1 (43,6)	0,995 (43,6)	0,591 (43,6)

**Taula 6.1 – Taula de resultats amb nivell de senyal = 75 dB, SNR = 0 dB, i rang de freqüències de 300 Hz a 10 KHz**

A la *taula 6.1* es poden analitzar els resultats que es donen pels diferents índexs. Amb els resultats a la mà, es pot dir que *l'Articulation Index* només té en compte el nivell de les diferents bandes de freqüència, per tant, totes les millores aconseguen una millora (encara que sigui mínima) respecte del senyal original. Només sabent com funciona el càlcul d'aquest mètode es pot dir que a més amplificació s'aconsegueixi a les bandes, millor resultat donarà, això fa que evidentment, per aquest cas, la millora que obté un millor resultat és la que s'aplica la millora amb major amplificació, encara que també hem de

tenir en compte que aplica una amplificació total de quasi 33 dB, un valor que podria distorsionar el senyal, i per tant, empitjorar la intel·ligibilitat. En verd es poden veure marcades les millores que assolixen un millor resultat pel mètode de *l'Articulation Index*.

El cas del STOI és el que obté uns nivells d'intel·ligibilitat més alts, i després de reproduir els senyals, sembla que és el que millor correla amb la realitat. En aquests cas, com en l'anterior, on s'aplica una major amplificació s'aconsegueix millor nivell, però no necessitem unes gran amplificacions per obtenir uns nivells que indiquin que el senyal és totalment intel·ligible. Marcats de color taronja es poden observar els diferents valors a partir dels quals es podria considerar que el senyal es pot entendre perfectament, i com s'observa, amb amplificacions de només entre 4 i 6 dB, ja s'aconsegueix una millora adequada.

Observant els valors que dona el mètode SII, es pot veure clarament com a més a més de la diferència de nivell entre el senyal de veu i el soroll de fons es té en compte el nivell del senyal de veu. A la *taula 6.1* tenim marcats en vermell els valors que tenen una excessiva amplificació i això fa que el nivell de la intel·ligibilitat baixi. Això demostra el que es va veure a *l'apartat 3.2*, allà s'explicava que aquest mètode tenia en compte la possible distorsió a causa d'un excessiu nivell del senyal, per tant, quan l'amplificació és massa elevada, el nivell d'intel·ligibilitat baixa.

			Ambient aeroport			Soroll blanc			Soroll rosa		
			Diferència mínima	Factor amplificació	AI	STOI	SII	AI	STOI	SII	AI
<b>Sense millora</b>	-	-	0	0,51	0,007	0,033	0,444	0,051	0	0,381	0,007
<b>Millora 1</b>	<b>0</b>	-	0,007 (5,7)	0,623 (5,7)	0,86 (5,7)	0,145 (7,1)	0,514 (7,1)	0,125 (7,1)	0,06 (5,9)	0,458 (5,9)	0,072 (5,9)
	<b>5</b>	-	0,06 (10,3)	0,696 (10,3)	0,234 (10,3)	0,223 (11,8)	0,614 (11,8)	0,256 (11,8)	0,121 (10,5)	0,53 (10,5)	0,199 (10,5)
	<b>10</b>	-	0,182 (15,2)	0,779 (15,2)	0,373 (15,2)	0,322 (16,8)	0,727 (16,8)	0,378 (16,8)	0,208 (15,4)	0,64 (15,4)	0,317 (15,4)
	<b>15</b>	-	0,347 (20,2)	0,846 (20,2)	0,503 (20,2)	0,428 (21,7)	0,812 (21,7)	0,49 (21,7)	0,316 (20,4)	0,729 (20,4)	0,426 (20,4)
	<b>20</b>	-	0,514 (25,1)	0,885 (25,1)	0,623 (25,1)	0,533 (26,7)	0,872 (26,7)	0,59 (26,7)	0,449 (25,4)	0,79 (25,4)	0,525 (25,4)
<b>Millora 2</b>	-	<b>Amplificació màxima</b>	0,814 (34,8)	0,913 (34,8)	0,68 (34,8)	0,845 (44,8)	0,969 (44,8)	0,59 (44,8)	0,683 (36,8)	0,836 (36,8)	0,537 (36,8)
<b>Millora 3</b>	-	<b>0,2</b>	0,012 (7)	0,564 (7)	0,049 (7)	0,114 (7,2)	0,517 (7,2)	0,116 (7,2)	0,066 (9)	0,415 (9)	0,061 (9)
	-	<b>0,4</b>	0,174 (15,7)	0,664 (15,7)	0,166 (15,7)	0,259 (16,3)	0,657 (16,3)	0,271 (16,3)	0,258 (20,8)	0,604 (20,8)	0,249 (20,8)
	-	<b>0,6</b>	0,464 (24,4)	0,782 (24,4)	0,343 (24,4)	0,453 (25,3)	0,805 (25,3)	0,467 (25,3)	0,553 (31,8)	0,777 (31,8)	0,461 (31,8)
	-	<b>0,8</b>	0,746 (32,8)	0,87 (32,8)	0,523 (32,8)	0,664 (34,7)	0,913 (34,7)	0,618 (34,7)	0,83 (43,6)	0,899 (43,6)	0,598 (43,6)

**Taula 6.2 – Taula de resultats amb nivell de senyal = 75 dB, SNR = -10 dB, i rang de freqüències de 300 Hz a 10 KHz**

A la *taula 6.2* es poden observar els resultats de les millores en el cas que el senyal de veu tingui un nivell de 10 dB inferior al senyal de soroll de fons. Amb això es simula un exemple d'un lloc molt sorollós, en el qual soroll de fons seria molt alt respecte al senyal de veu.

Amb el mètode de l'*Articulation Index* es pot veure com per aquest cas, si es vol aconseguir un valor alt, és necessària una amplificació molt alta, ja que el soroll està al mateix nivell que en el cas anterior, però el senyal de veu està molt per sota. En taronja estan marcats el resultats que donen una bona



intel·ligibilitat, i com podem veure són amplificacions de més de 30 dB al senyal de veu original.

En el cas del STOI, sí que hi ha millores que indiquen un bon valor d'intel·ligibilitat. Amb aquest índex es poden veure en color verd senyalades les millores que aconseguen un bon nivell d'intel·ligibilitat. A la taula es pot veure que amb una amplificació de 15 dB aplicant la primera millora, ja es podria aconseguir un senyal acceptable. Amb amplificacions més grans, fins i tot arriben a donar valors que es consideren d'una intel·ligibilitat perfecta.

En el cas del SII és el més difícil de que doni un valor acceptable per a la intel·ligibilitat de la paraula. En blau es poden veure marcats a la taula els resultats que estan considerats com una intel·ligibilitat excel·lent (valors majors de 0.75 com es va poder veure a la *figura 3.3*). Marcats de color groc es poden observar les millores que es considera que donen una bona intel·ligibilitat (valors entre 0,6 i 0,75).

Amb aquesta taula es pot veure que en els casos en què el senyal de veu estigui molt per baix del soroll, es necessiten grans amplificacions per poder assolir uns resultats acceptables.

			Ambient aeroport			Soroll blanc			Soroll rosa		
	Diferència mínima	Factor amplificació	AI	STOI	SII	AI	STOI	SII	AI	STOI	SII
<b>Sense millora</b>	-	-	0,274	0,734	0,34	0,3	0,68	0,31	0,215	0,626	0,224
<b>Millora 1</b>	<b>0</b>	-	0,298 (1,7)	0,68 (1,7)	0,464 (1,7)	0,372 (1,9)	0,653 (1,9)	0,472 (1,9)	0,269 (2,7)	0,582 (2,7)	0,401 (2,7)
	<b>5</b>	-	0,465 (6,7)	0,728 (6,7)	0,602 (6,7)	0,482 (6,9)	0,704 (6,9)	0,574 (6,9)	0,4 (7,7)	0,645 (7,7)	0,519 (7,7)
	<b>10</b>	-	0,631 (11,7)	0,758 (11,7)	0,725 (11,7)	0,595 (11,9)	0,739 (11,9)	0,662 (11,9)	0,555 (12,7)	0,685 (12,7)	0,618 (12,7)
	<b>15</b>	-	0,796 (16,7)	0,777 (16,7)	0,818 (16,7)	0,71 (16,9)	0,763 (16,9)	0,732 (16,9)	0,689 (17,7)	0,71 (17,7)	0,695 (17,7)
	<b>20</b>	-	0,928 (21,7)	0,79 (21,7)	0,833 (21,7)	0,813 (21,9)	0,781 (21,9)	0,732 (21,9)	0,803 (22,7)	0,727 (22,7)	0,702 (22,7)
<b>Millora 2</b>	-	<b>Amplificació màxima</b>	1 (27)	0,917 (27)	0,806 (27)	0,975 (37)	0,95 (37)	0,671 (37)	0,918 (29)	0,92 (29)	0,666 (29)
<b>Millora 3</b>	-	<b>0,2</b>	0,569 (8,9)	0,847 (8,9)	0,551 (8,9)	0,503 (8,8)	0,815 (8,8)	0,497 (8,8)	0,512 (11)	0,796 (11)	0,463 (11)
	-	<b>0,4</b>	0,857 (17,8)	0,922 (17,8)	0,714 (17,8)	0,715 (17,7)	0,937 (17,7)	0,658 (17,7)	0,817 (23)	0,895 (23)	0,644 (23)
	-	<b>0,6</b>	0,999 (26,6)	0,982 (26,6)	0,795 (26,6)	0,883 (26,6)	0,994 (26,6)	0,696 (26,6)	0,971 (34)	0,933 (34)	0,655 (34)
	-	<b>0,8</b>	1 (35,3)	0,998 (35,3)	0,791 (35,5)	0,977 (35,7)	1 (35,7)	0,651 (35,7)	1 (45,9)	0,969 (45,9)	0,588 (45,9)

**Taula 6.3 – Taula de resultats amb nivell de senyal = 75 dB, SNR = 0 dB, i rang de freqüències de 0 Hz a 20 KHz**

A la *taula 6.3* es pot observar com utilitzar tot el rang de freqüències audibles no és la forma més òptima d'aplicar les millores. Degut al que es va poder veure a l'*apartat 2.4*, la veu no es troba a tot el rang, sinó que només es troba a una part de l'espectre freqüencial. Per això, aplicar una amplificació a les freqüències baixes i a les altes, fa que estiguem aplicant una amplificació molt més gran per tal d'aconseguir uns resultats iguals a aplicar la millora només al rang on la veu conté la majoria de la informació.

A més a més, es pot observar que d'aquesta forma no només necessites una major amplificació, sinó que els resultats obtinguts són pitjors que per al cas on només s'utilitza el rang útil de la veu.

Diferència mínima		Ambient aeroport			Soroll blanc			Soroll rosa		
		AI	STOI	SII	AI	STOI	SII	AI	STOI	SII
-	<b>Sense Millora</b>	0,274	0,734	0,34	0,3	0,68	0,31	0,215	0,626	0,224
0	<b>Millora 1</b>	0,295 (0,3)	0,787 (0,3)	0,463 (0,3)	0,372 (1,4)	0,784 (1,4)	0,472 (1,4)	0,267 (0,9)	0,75 (0,9)	0,4 (0,9)
	<b>Millora 2</b>	0,303 (0,3)	0,726 (0,3)	0,368 (0,3)	0,339 (1,4)	0,738 (1,4)	0,341 (1,4)	0,234 (0,9)	0,639 (0,9)	0,245 (0,9)
5	<b>Millora 1</b>	0,449 (4,1)	0,884 (4,1)	0,598 (4,1)	0,478 (5,4)	0,9 (5,4)	0,575 (5,4)	0,393 (4,8)	0,865 (4,8)	0,516 (4,8)
	<b>Millora 2</b>	0,457 (4,1)	0,843 (4,1)	0,509 (4,1)	0,442 (5,4)	0,827 (5,4)	0,43 (5,4)	0,347 (4,8)	0,76 (4,8)	0,349 (4,8)
10	<b>Millora 1</b>	0,611 (8,6)	0,956 (8,6)	0,717 (8,6)	0,588 (10,1)	0,956 (10,1)	0,663 (10,1)	0,545 (9,3)	0,928 (9,3)	0,613 (9,3)
	<b>Millora 2</b>	0,612 (8,6)	0,919 (8,6)	0,631 (8,6)	0,552 (10,1)	0,884 (10,1)	0,527 (10,1)	0,491 (9,3)	0,838 (9,3)	0,457 (9,3)
15	<b>Millora 1</b>	0,775 (13,4)	0,971 (13,4)	0,81 (13,4)	0,704 (14,9)	0,979 (14,9)	0,734 (14,9)	0,68 (14,2)	0,956 (14,2)	0,691 (14,2)
	<b>Millora 2</b>	0,783 (13,4)	0,953 (13,4)	0,728 (13,4)	0,668 (14,9)	0,934 (14,9)	0,61 (14,9)	0,637 (14,2)	0,898 (14,2)	0,549 (14,2)
20	<b>Millora 1</b>	0,913 (18,4)	0,976 (18,4)	0,827 (18,4)	0,807 (19,9)	0,99 (19,9)	0,735 (19,9)	0,8 (19,2)	0,97 (19,2)	0,7 (19,2)
	<b>Millora 2</b>	0,919 (18,4)	0,97 (18,4)	0,789 (18,4)	0,775 (19,9)	0,97 (19,9)	0,655 (19,9)	0,758 (19,2)	0,94 (19,2)	0,622 (19,2)

Taula 6.4 – Taula de comparació de resultats de les millores 1 i 2 amb nivell de senyal = 75 dB, SNR = 0 dB, i rang de freqüències de 300 Hz a 10 KHz

A la *taula 6.4* es pot observar la comparació entre les millores 1 i 2 amb diferents paràmetres. A l'*apartat 5.4* ja es va explicar com funcionava cadascuna d'aquestes millores, la primera millora es basava en les diferències de nivell entre el senyal de veu i el soroll i la segona millora només consistia en pujar el nivell de forma constant en cadascuna de les bandes. El funcionament d'aquesta comparació es molt simple, primer de tot s'aplica la millora 1, i tot seguit, s'aplica la millora 2 amb una amplificació total igual a la de la millora 1. D'aquesta manera els dos senyals amplificats tenen la mateixa potència i les seves intel·ligibilitats es poden comparar objectivament.

A la taula es pot veure el que esperàvem des d'un primer moment, que la primera millora obtindria uns resultats millors als de la segona millora. Gràcies a què aquesta primera millora té en compte les diferències de nivell entre el senyal de veu i el de soroll, optimitza millor a les bandes on s'ha d'amplificar el senyal original.

Els casos del AI i SII, com es va explicar al *capítol 3*, basen el seu càlcul exactament en les diferències de nivell a les diferents bandes de freqüència, així que amb aquests mètodes era segur que la primera millora obtindria millors resultats, especialment el mètode SII que fa el càlcul amb bandes de terç d'octava, de la mateixa manera que la millora. Exactament on es demostra que la primera millora és molt més òptima que la segona és en els resultats del mètode STOI, que tot i basar-se en un altre mètode per fer el càlcul del índex, també aconsegueix uns valors superiors per la primera millora.

Factor amplificació		Ambient aeroport			Soroll blanc			Soroll rosa		
		AI	STOI	SII	AI	STOI	SII	AI	STOI	SII
-	Sense Millora	0,274	0,734	0,34	0,3	0,68	0,31	0,215	0,626	0,224
0,2	Millora 3	0,548 (7)	0,9 (7)	0,545 (7)	0,49 (7,2)	0,856 (7,2)	0,499 (7,2)	0,491 (9)	0,841 (9)	0,457 (9)
	Millora 2	0,554 (7)	0,891 (7)	0,585 (7)	0,486 (7,2)	0,849 (7,2)	0,471 (7,2)	0,491 (9)	0,838 (9)	0,457 (9)
0,4	Millora 3	0,836 (15,7)	0,957 (15,7)	0,71 (15,7)	0,71 (16,3)	0,956 (16,3)	0,66 (16,3)	0,801 (20,8)	0,951 (20,8)	0,643 (20,8)
	Millora 2	0,845 (15,7)	0,96 (15,7)	0,758 (15,7)	0,69 (16,3)	0,943 (16,3)	0,624 (16,3)	0,801 (20,8)	0,951 (20,8)	0,643 (20,8)
0,6	Millora 3	0,997 (24,4)	0,975 (24,4)	0,792 (24,4)	0,878 (25,3)	0,993 (25,3)	0,699 (25,3)	0,965 (31,8)	0,982 (31,8)	0,658 (31,8)
	Millora 2	0,999 (24,4)	0,978 (24,4)	0,804 (24,4)	0,877 (25,3)	0,99 (25,3)	0,686 (25,3)	0,965 (31,8)	0,979 (31,8)	0,658 (31,8)
0,8	Millora 3	1 (32,8)	0,986 (32,8)	0,793 (32,8)	0,975 (34,7)	0,999 (34,7)	0,654 (34,7)	1 (43,6)	0,995 (43,6)	0,591 (43,6)
	Millora 2	1 (32,8)	0,984 (32,8)	0,766 (32,8)	0,97 (34,7)	0,998 (34,7)	0,674 (34,7)	1 (43,6)	0,991 (43,6)	0,591 (43,6)

Taula 6.5 – Taula de comparació de resultats de les millores 2 i 3 amb nivell de senyal = 75 dB, SNR = 0 dB, i rang de freqüències de 300 Hz a 10 KHz

A la *taula 6.5* es poden observar els resultats de la comparació entre les millores 2 i 3. Com es va explicar a l'*apartat 5.4*, la segona millora només té en compte el senyal de soroll de fons, i el que fa és aplicar una amplificació al senyal de veu dependent de les potències de les diferents bandes per el senyal de soroll.

El primer que es pot veure, marcat de color verd a la taula, és que els resultats de les millores 2 i 3 pel cas del soroll rosa són pràcticament iguals. Això és

evident, ja que el soroll rosa té un espectre pla i fa que tenint només en compte l'espectre del soroll, les dues millores siguin exactament les mateixes. Les úniques petites diferències que trobem són que en fer l'anàlisi del senyal de soroll rosa el càlcul no surt perfectament pla i quan es fan gran amplifcacions aquestes petites diferències es fan una mica més grans.

Una vegada descartat el soroll rosa ja que fa exactament el mateix, pels altres dos sorolls de fons els dos mètodes tenen un comportament bastant semblant i és difícil treure conclusions de quin dels dos mètodes funciona millor, per uns casos funciona millor el segon mètode i per altres el tercer.

D'aquesta manera podem arribar a la conclusió que la millora més idònia es la primera, ja que tenint en compte tant el senyal de veu com el senyal de soroll, es pot aplicar un filtratge més òptim i obtenir millors resultats amb una amplifcació total menor.

## 7. Conclusions

Durant aquest projecte s'ha creat un programa que permet aplicar millores a un senyal de veu per tal d'incrementar la intel·ligibilitat de la paraula. Amb això aconseguim simular com es podria millorar un senyal de veu per tal que fos més comprensible en un lloc públic. A més a més, aquest programa dona la possibilitat de quantificar objectivament i subjectivament els resultats obtinguts amb les millores, mitjançant el càlcul dels diferents índexs que podem triar i mitjançant els reproductors.

Per la realització d'aquest projecte s'ha posat en pràctica coneixements que s'han obtingut durant el grau, especialment d'acústica i processament d'àudio. S'ha fet un estudi de la producció del senyal de veu i del comportament del sistema auditiu humà per tal d'entendre com s'havien d'aplicar les diferents millores, i el perquè dels mètodes que utilitzen els diferents índexs d'intel·ligibilitat per realitzar els càlculs.

A més a més de posar en pràctica els coneixements que s'han après, també he ampliat els coneixements de programació amb Matlab, ja que a partir d'aquest programa s'ha realitzat tot el programa. Especialment s'han adquirit grans coneixements de la llibreria *DSP System Toolbox*.

El programa que s'ha creat permet aplicar tres millores diferents al senyal de veu i visualitzar d'una manera clara i senzilla els resultats d'aquestes millores. Per la realització del programa ens vam haver de centrar en dues parts molt diferenciades, una primera part en què es van implementar els diferents mètodes pel càlcul de la intel·ligibilitat i es va haver d'entendre com funcionava cadascun dels mètodes per tal de poder utilitzar els mètodes que ja estaven implementats, però que necessitaven que se'ls hi enviessin unes dades específiques. Durant la segona part del projecte ens vam centrar més en la aplicació i comparació de les diferents millores.

Com a conclusió, es pot assegurar que la millora 1 es preferible de les altres, el fet de que tingui en compte el senyal de veu i el senyal de soroll, fa que es puguin optimitzar de millor manera les diferències entre els dos senyals i s'obtinguin uns millors resultats. Després d'analitzar tots els resultats, també es

pot arribar a la conclusió de quel índex d'intel·ligibilitat més fiable és el STOI, ja que després de veure els valors dels resultats i reproduint els senyals, es el mètode que més s'apropa a la realitat

## **7.1 Futures implementacions**

L'estructura del programa deixa oberta la possibilitat de poder fer futures amplificacions als programa.

Durant el projecte hem parlat d'altres mètodes com per exemple el ALcons o el STI, inclosos els seus germans petits. Amb una possible futura ampliació es podria considerar un local específic i utilitzar aquest sistema per tal de calcular la intel·ligibilitat.

Amb aquest cas del STI seria bàsic tenir en compte que no es podria tractar d'una forma tan general com els que hem implementat en aquest projecte, ja que son necessaris càlculs com el temps de reverberació, per tant es podria centrar més en el tema de l'acústica arquitectònica, de la qual es va parlar a l'apartat 2.5, ja que durant la implementació aquest programa no la hem tingut en compte.

A més a més de les millores ja implementades, es podrien implementar unes altres millores com ara jugar amb la velocitat de la senyal de veu, ja que en llocs sorollosos el ser humà tendeix a, a més a més de parlar més alt, reduir la velocitat del diàleg per tal de intentar fer-ho més comprensible. Un altre millora possible seria un compressor que actues de forma ràpida per tal de millorar la intel·ligibilitat, especialment per comparar-la amb la taula 6.2, ja que aquest mètode es molt útil per a llocs molt sorollosos.



## 8. Bibliografía

Lawrence E. Kinsler. *Fundamental of Acoustics*. 3ª edición. John Wiley & Sons, 1982.

Leo L. Beranek. *Acoustics*. 2ª edición. American Institute of Physics, 1986.

T. Houtgast and H.J.M. Steeneken, "A review of the MTF concept in room acoustics and its use for estimating speech intelligibility in auditoria"

T. Houtgast and H.J.M. Steeneken, "A physical method for measuring speech-transmission quality"

Karl D. Kryter, "Methods for the Calculation and Use of the Articulation Index"

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen, "An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech"

Koenraad S. Rhebergen and Niek J. Versfeld "A Speech Intelligibility Index-based approach to predict the speech reception threshold for sentences in fluctuating noise for normal-hearing listeners"

Ole-Herman Bjør, "Measure Speech Intelligibility with a Sound Level Meter"

Agustín Martín Domingo, *Apuntes de Acústica*. 2ª edición. 2013

Matlab, *Creating Graphical User Interfaces*

MathWorks. Help. [mathworks.com](https://www.mathworks.com/help)

[stackoverflow.com](https://stackoverflow.com)

# 9. ANNEX A - CODI

## ANNEX A.1 – Funcions

### A.1.1 – Funcions pel càlcul de la intel·ligibilitat

#### Articulation Index

```
function A = AI(x,n,sn,pot)

[Hd, Nfc] = filtroAI();

R = zeros(1,20);

for k=1:Nfc
    hq(k) = convert(Hd(k),'df1');
    y = filter(hq(k),x);
    yn = filter(hq(k),n);

    power(k) = rms(y); % calcul de la potencia de la senyal per
cada banda
    dbpow(k) = 20*log10(power(k)); % calcul de la potencia en dB

    powern(k) = rms(yn); % calcul de la potencia del soroll de
fons per cada banda
    dbpown(k) = 20*log10(powern(k));

    % Atenuació o amplificació del soroll dependent del SNR
    if sn>0
        Be(k) = dbpown(k)-sn;
    else
        Be(k) = dbpown(k)+abs(sn);
    end

    V(k) = dbpow(k);
end

Be = Be+pot; % suma de la potencia de veu als dos senyals
V = V+pot;

A = artind(V,R,Be,'kryter');
end

% ----- function filter -----
function [hd, Nfc] = filtroAI()

frecuencyMatrix = [200 330; 330 430; 430 560; 560 700; 700 840; ...
                    840 1000; 1000 1150; 1150 1310; 1310 1480; ...
                    1480 1660; 1660 1830; 1830 2020; 2020 2240; ...
                    2240 2500; 2500 2820; 2820 3200; 3200 3650; ...
```

```

3650 4250; 4250 5050; 5050 6100];

Nfc = size(frequencyMatrix,1)

for i=1:Nfc

f=fdesign.bandpass('N,Fc1,Fc2',4,frequencyMatrix(i,1),frequencyMatrix(
i,2),44100);
    hd(i) = design(f,'butter');
end

end

% ----- function artind -----
function [AI,W,f]=artind(V,R,Be,type)
%
% [AI,W,f]=artind(V,R,Be,type)
%
% Estimates the articulation index, AI, based
% on the paper by French and Steinberg (1946).All inputs
% are 1x20 vectors with values corresponding to the average of the
% 20 frequency bands given below.
%
% V is "the speech level of the talker at two inches
% from the lips, as determined by a sound level measurement
% using 40-dB weighting."
%
% R is the orthotelephonic response of the communication system
% at the appropriate frequencies.
%
% Be represents noise intensity per frequency level of the total
% noise from all sources except that produced by the received speech.
%
% 'type' is a string defining which AI calculation to use. Set:
%
% 'french'           from French Steinberg's Method (JASA, 1949),
default
% 'kryter'           from Kryter's Method (JASA, 1962)
% 'dot_nonsense_syl' from Mueller & Killion (1990) used to measure
AI based on human pure
%                   tone thresholds (V). Enter empty values for
othe parameters. V is
%                   the pure tone threshold for frequencies 250
300 400 500 600 800 1000 1250 1750 2000 2500 3500 4000 5500
%
% 'Ao'               from Pavlovic (1991). Estimates persons AI
based on pure tone thresholds for frequencies:
%                   500 1000 2000 4000. V is the persons thresholds
and those respective frequencies. Enter empty
%                   brackets for the other parameters.
%
% A is the estimated articulation index. A is between 0 (0%
intelligibility and
% 1 (100% intelligibility).
%
% W is the weighting function for each band. Value of 1 means band is
contributing
% maximally to speech intelligibility. Value of 0 means band is not
contributing

```

```

% at all.
%
% f is the frequency centers from which W was calculated.
%
% for the kryter method the center frequencies and their minmax ranges
are (in Hz):
%       270           200           330
%       380           330           430
%       490           430           560
%       630           560           700
%       770           700           840
%       920           840           1000
%      1070          1000          1150
%      1230          1150          1310
%      1400          1310          1480
%      1570          1480          1660
%      1740          1660          1830
%      1920          1830          2020
%      2130          2020          2240
%      2370          2240          2500
%      2660          2500          2820
%      3000          2820          3200
%      3400          3200          3650
%      3950          3650          4250
%      4650          4250          5050
%      5600          5050          6100
%
%
%
%
% Written by Ikaro Silva 2005

```

```

if(isempty(type) | strcmp(type,'french'))

```

```

%%
% The 20 band centers and ranges (Hz):
% 310    250-375
% 440    375-505
% 575    505-645
% 720    645-795
% 875    795-955
% 1040   955-1130
% 1225   1130-1315
% 1415   1315-1515
% 1615   1515-1720
% 1825   1720-1930
% 2035   1930-2140
% 2250   2140-2355
% 2475   2355-2600
% 2750   2600-2900
% 3080   2900-3255
% 3470   3255-3680
% 3940   3680-4200
% 4530   4200-4860
% 5300   4860-5720
% 6350   5720-7000

```

```

%

%%Defining Parameters and Tables%%
W=zeros(1,20);
p=12;

%Value of Bs' according to table VI in dB vs 10^-16 watt/cm^2
Bsp=[36.5 36.6 35.7 33.4 30.7 28.3 26 24 22.1 20.4 18.9 17.5 ...
     16.1 14.6 13 11.3 9.5 7.5 5.1 2.5];

%X= bo -K
X=-1*[1.5 8.0 11.6 14.1 15.7 16.7 17.5 18.3 19.4 21 23.3 ...
     25.2 26.6 27.8 28.5 28.9 28.8 27.8 25.1 19.7];

%The number in dB that the noise produced in any band, by speech
in any
%lower band, is below the long average intensity of the speech in
the lower
%band from table VIII

Q=[49 62 70 75 78 81 83 85 85 85 86 86 87 87 87 87 87 87 87;...
   0 44 56 64 70 74 78 80 82 83 84 85 85 86 86 87 87 87 87;...
   0 0 42 52 61 67 71 75 78 79 81 83 84 85 85 86 86 87 87;...
   0 0 0 40 50 57 64 68 71 75 78 79 81 82 83 84 85 86 87;...
   0 0 0 0 38 49 56 62 66 70 72 75 78 79 81 83 85 85 86;...
   0 0 0 0 0 36 46 54 59 64 67 70 72 75 77 80 82 84 85;...
   0 0 0 0 0 0 35 45 51 56 61 65 69 72 75 77 79 81 84;...
   0 0 0 0 0 0 0 35 43 50 55 59 64 67 71 74 77 80 83;...
   0 0 0 0 0 0 0 0 35 42 47 52 57 63 67 71 74 78 81;...
   0 0 0 0 0 0 0 0 0 34 41 47 52 57 63 67 71 76 79;...
   0 0 0 0 0 0 0 0 0 0 33 40 46 52 57 63 68 72 77;...
   0 0 0 0 0 0 0 0 0 0 33 40 47 54 60 65 71 75;...
   0 0 0 0 0 0 0 0 0 0 34 41 50 55 61 67 72;...
   0 0 0 0 0 0 0 0 0 0 34 42 49 56 64 70;...
   0 0 0 0 0 0 0 0 0 0 35 43 51 59 65;...
   0 0 0 0 0 0 0 0 0 0 35 45 54 61;...
   0 0 0 0 0 0 0 0 0 0 35 46 56;...
   0 0 0 0 0 0 0 0 0 0 36 49;...
   0 0 0 0 0 0 0 0 0 0 39];

%Values of (B(+X)-X as a function of B-X. Table VII
BpXmX=[[-9:1:9]' [0.5 0.6 0.8 1 1.2 1.5 1.8 2.1 2.5 3.0 3.5 4.1
4.8 5.5 6.2 7.0 7.8 8.6 9.5]'];

%frequency center/range of each band. In case of plotting f vs W.
f=[310 440 575 720 875 1040 1225 1415 1615 1825 2035 2250 2475
2750 3080 3470 3940 4530 5300 6350];

%Values of m as a function of Z to the nearest dB (in case of high
level
%noises from Table II:
mZ=[[54:91]'
[ones(7,1);ones(5,1)*2;ones(5,1)*3;ones(4,1)*4;ones(3,1)*5;ones(3,1)*6
;ones(3,1)*7;ones(3,1)*8;ones(3,1)*9;ones(2,1)*10]];

%%Calculation of AI%%

```

```

    Bs=Bsp + (V-90) + R; % level in dB vs 10^-16 watt/cm^2 of the
long average intensity per cycle
                                % of the received speech (intensity being
expressed
                                % as free field intensity

    %Interband masking of speech, produced in band n by speech in band
k
    %So n is from 2-20 and k is from 1-19. Bnk only applies to band 2-
20.
    Bmask=triu(repmat(Bs(1:19),[19 1]));
    Bnk=Bmask - Q;
    Bnk(Bnk==0)=NaN;

    %Combine masking in a power basis (eq. 21) to obtain total inter-
speech
    %masking for respective band. Append 0 for the first band
    Bn=[0 nansum(10.^(Bnk./10))]; %add on a power basis
    Bn=10*log10(Bn);

    Bf= Bs-24; % Self masking noise
    B=10.^(Be/10)+ 10.^(Bf/10) + 10.^(Bn/10); %add on a power basis
    B=10*log10(B);

    Z=B-X; %effective level (level above threshold of a critical band
of noise)
    z=round(Z);
    % if(any(z>9) | any(z<-9))
    %     error('Effective level Z out of range.')
    % end
    %Calculate m using mZ table
    m=zeros(1,20);
    ind=find(Z>54);

    for i=1:length(ind),
        [val,trash]=find(mZ==round(Z));
        m(ind(i))=mZ(val,2);
    end

    BpX=zeros(20,1);
    for i=1:20,
        [ind,trash]=find(BpXmX(:,1)==z(i));
        BpX(i)=BpXmX(ind,2)+X(i); % (B(+)X)-X+X = B(+)X
    end

    W=(1/30).*[Bs+p-6-m-BpX'];
    AI=0.05*sum(W);

elseif(strcmp(type,'kryter'))

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Kryter's Method%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Not corrected for loudspeaker presentation of speech in
typical
    %semireverberant room

    f=[270 380 490 630 770 920 1070 1230 1400 1570 1740 1920 2130
2370 2660 3000 3400 3950 4650 5600]; %from Beranek

```

```

frange=[200 330 430 560 700 840 1000 1150 1310 1480 1660 1830
2020 2240 2500 2820 3200 3650 4250 5050]; %frequency range for each
band
frange=[frange' [frange(2:end) 6100]'];
df=diff(frange,[],2);
th=-1*[6 10 13 16 16 16 17 18 19 20 22 23 26 28 29 30 29 27 24
21];%threshold of sounds with continous spectra
Bsp=[50 50 48 45 43 40 37 34 32 31 29 28 26 25 24 23 21 20 19
19]; %ideal speech spectrum
Msk=[[80:150]' floor([0:0.2:14]')]; %Correction for non-linear
growth of masking
max_speech=[114 110 107 105 103 101 99 97 97 97 97 97 97 97 97
97 98 100 103 104];
%upward spread of masking
high_lev=[46:10:96]';
high_mask=zeros(6,2,5);
high_mask(:,:,1)=[50 75 150 200 200 250]' [45 35 25 20 15
10]';
high_mask(:,:,2)=[100 150 250 400 500 500]' [40 30 23 18 13
8]';
high_mask(:,:,3)=[200 300 500 800 1000 1000]' [35 25 20 15 10
5]';
high_mask(:,:,4)=[200 500 1000 1500 1500 1500]' [40 25 15 10
5 3]';
high_mask(:,:,5)=[200 800 2000 3000 3000 3000]' [40 20 5 0 0
0]';
high_ind=[0 800 1600 2400 3200 5200]; %frequency upper bound
values

%Step 1
Bs=Bsp+R;

%RMS of input speech spectrum
Vrms=10*log10(10.^(V/10)*df);
Speech_Peaks= Bs + (Vrms-65);

%Step 2
Band_SL=Be - th;
noise_spec=Band_SL;
ind=find(Band_SL >80);
for i=1:length(ind),
[val,trash]=find(Msk==round(Band_SL(ind(i))));
noise_spec(ind(i))=noise_spec(ind(i))+ Msk(val,2);
end

%Step 3
noise_spec=noise_spec+th;
[val,ind]=max(noise_spec);
n=noise_spec;
n(ind)=-inf;
[st_pt]=find(n >= (val-3));
if isempty(st_pt)
st_pt=20;
else
st_pt=st_pt(end);
end

low_m=zeros(1,st_pt);

```

```

Lev=noise_spec(st_pt);
if(Lev>46),
    %Downward spread of masking
    for i=st_pt:-1:1,

        %Upward line to the left of st_pt with 10dB per octave
slope
        low_m(i)= (Lev-57) + (10/log(2))*log(f(st_pt)/f(i));
    end
    tLev=Lev;
    if(tLev>95)
        tLev=95;
    end
    %low_m(end)=[];

    [lev_ind]=find(high_lev>tLev)-1;
    [f_ind]=find(high_ind>f(st_pt-1))-1; % cambio a st_pt-1,
incluido -1
    width=high_mask(lev_ind(1),1,f_ind(1));
    slope=high_mask(lev_ind(1),2,f_ind(1));

    %Upward spread of masking
    high_m=zeros(1,20-st_pt);
    j=1;
    for i=st_pt+1:20, %incluido el +1 posteriormente

        if(f(i)< (st_pt+width)),
            high_m(j)=Lev;
        else
            high_m(j)= Lev +
(slope/log(2))*log(f(st_pt)/f(i));
        end
        j=j+1;
    end
    else
        high_m=zeros(1,20-st_pt);
        low_m=zeros(1,st_pt);
    end

    %Step 4
    noise_spec=max(noise_spec,[low_m high_m]);
    noise_spec=max(noise_spec,th);
    Speech_Peaks=min(Speech_Peaks,max_speech);
    W=Speech_Peaks - noise_spec;
    W(W<0)=0;
    W(W>30)=30;

    %Step 5
    AI=sum(W)/600;

elseif(strcmp(type,'Ao'))

    f=[500 1000 2000 4000];
    W=max(V,20);
    W=min(W,50);
    A=sum(50-W)/120;

elseif(strcmp(type,'dot_nonsense_syl'))

```



```

f=[250 300 400 500 600 800 1000 1250 1750 2000 2500 3500 4000
5500];
N=length(f);
dot=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 1;...
%15HL
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;...
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1;...
1 0 0 1 0 0 1 1 1 1 1 1 1 1 0;...
0 0 0 0 0 0 1 1 1 1 1 1 1 0 1;...
0 1 1 0 1 1 0 0 1 1 1 1 1 0 0;...
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1;...
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;...
1 0 1 0 0 0 1 1 1 1 1 1 1 1 1;...
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0;...
0 1 0 1 0 0 1 1 1 1 1 1 1 1 1;...
0 0 0 0 1 1 0 1 0 0 0 0 0 0 0;...
0 0 1 0 0 0 1 1 1 1 1 1 1 1 0;...
0 0 0 1 0 0 0 0 1 1 1 1 1 1 0;...
0 0 0 0 1 1 1 1 1 1 1 1 1 0 0;...
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0];%48 HL

```

```

rows=linspace(15,48,18)';
[ddots,dth]=meshgrid(rows,V);
D=abs(ddots-dth);
[trash,c1]=min(D,[],2);
W=zeros(1,N);
for i=1:N,
    W(i)=sum(dot(c1(i):end,i));
end

A=sum(W(:))/100;

```

```

end
end

```

## SII

```

function Sn = SII_1(x,n,sn,pot)

[Hd Nfc] = filtroSII();

Tn = zeros(1,18);

for k=1:Nfc
    hq(k) = convert(Hd(k),'df1');
    y = filter(hq(k),x);
    yn = filter(hq(k),n);

    power(k) = rms(y);
    dbpow(k) = 20*log10(power(k));

    powern(k) = rms(yn);
    dbpown(k) = 20*log10(powern(k));

```

```

        if sn>0
            Nn(k) = dbpown(k)-sn;
        else
            Nn(k) = dbpown(k)+abs(sn);
        end
        En(k) = dbpow(k);
    end

    En = En+pot;
    Nn = Nn+pot;

    Sn = SII('E',En,'N',Nn);

end

% ----- Filtro SII -----
function [hd Nfc] = filtroSII()

frequencyMatrix = [141 178; 178 224; 224 282; ...
                  282 355; 355 447; 447 562; 562 708; ...
                  708 891; 891 1122; 1122 1413; 1413 1778; ...
                  1778 2239; 2239 2818; 2818 3548; 3548 4467; ...
                  4467 5623; 5263 7079; 7079 8913];

Nfc = size(frequencyMatrix,1)

for i=1:Nfc

f=fdesign.bandpass('N,Fc1,Fc2',4,frequencyMatrix(i,1),frequencyMatrix(
i,2),44100);
    hd(i) = design(f,'butter');
end

%hfvf = fvtool(hd,'FrequencyScale','log','color','white');
%title('1/3-Octave-Band Filter Bank')

end

% ----- Funció SII -----
function S = SII(varargin)

% "Methods for calculation of the Speech Intelligibility Index"
% (ANSI S3.5-1997)
%
% MATLAB implementation of Section 4.
%
% Note: The remaining sections of the standard, which provide means to
calculate input parameters required by
% the "core" SII procedure of Section 4, are implemented in separate
scripts:
% Section 5.1 in script Input_5p1.m "method based on the direct
measurement/ estimation of noise and speech spectrum levels at the
listener's position"
% Section 5.2 in script Input_5p2.m "method based on MTFI/CSNSL
measurements at the listener's position"

```

```

% Section 5.3 in script Input_5p3.m "method based on MTFI/CSNSL
measurements at the eardrum of the listener"
%
%
% Parameters are passed to the procedure through pairs of "identifier"
and corresponding "argument"
% Identifiers are always strings. Possible identifiers are:
%
%   'E' Equivalent Speech Spectrum Level (Section 3.6 in the
standard)
%   'N' Equivalent Noise Spectrum Level (Section 3.15 in the
standard)
%   'T' Equivalent Hearing Threshold Level [dBHL] (Section 3.23 in
the standard)
%   'I' Band Importance function (Section 3.1 in the standard)
%
% Except for 'E', which must be specified, all parameters are
optional. If an identifier is not specified a default value will be
used.
% Pairs of identifier and argument can occur in any order. However,
if an identifier is listed, it must be followed immediately by its
argument.
%
%
% Possible arguments for the identifiers are:
%
%   Arguments for 'E':
%       A row or column vector with 18 numbers stating the
Equivalent Speech Spectrum Levels in dB in bands 1 through 18.
%
%   Arguments for 'N':
%       A row or column vector with 18 numbers stating the
Equivalent Noise Spectrum Levels in dB in bands 1 through 18.
%       If this identifier is omitted, a default Equivalent Noise
Spectrum Level of -50 dB is assumed in all 18 bands (see note in
Section 4.2).
%
%   Arguments for 'T':
%       A row or column vector with 18 numbers stating the
Equivalent Hearing Threshold Levels in dBHL in bands 1 through 18.
%       If this identifier is omitted, a default Equivalent
Hearing Threshold Level of 0 dBHL is assumed in all 18 bands .
%
%   Arguments for 'I':
%       A scalar having a value of either 1, 2, 3, 4, 5, 6, or 7.
The Band-importance functions associated with each scalar are
%           1: Average speech as specified in Table 3
(DEFAULT)
%           2: various nonsense syllable tests where most
English phonemes occur equally often (as specified in Table B.2)
%           3: CID-22 (as specified in Table B.2)
%           4: NU6 (as specified in Table B.2)
%           5: Diagnostic Rhyme test (as specified in
Table B.2)
%           6: short passages of easy reading material
(as specified in Table B.2)
%           7: SPIN (as specified in Table B.2)
%
%
% The function returns the SII of the specified listening condition,
which is a value in the interval [0, 1].

```

```

%
%
% REMINDER OF DEFINITIONS & MEANINGS:
%
% Equivalent Speech Spectrum Level, E-prime
%       The SII calculation is based on free-field levels, even
though the quantity relevant for perception and intelligibility
%       is the level at the listener's eardrum.
%       The Equivalent Speech Spectrum Level is the speech
spectrum level at the center of the listener's
%       head (when the listener is temporarily absent) that
produces in an average human with unoccluded ears an eardrum speech
level equal
%       to the eardrum speech level actually present in the
listening situation to be analyzed.
%       Before the SII can be applied to a given listening
situation, the corresponding Equivalent Speech
%       Spectrum Level must be derived. For example, when speech
is presented over insert earphones (earphones inside the earcanal),
%       only the speech spectrum level at the eardrum is known.
Using the inverse of the freefield-to-eardrum transfer function (Table
3 of the standard)
%       this eardrum level must be "projected" into the freefield,
yielding the Equivalent Speech Spectrum Level.
%       Sections 5.1, 5.2, and 5.3 of the standard give three
examples of how to derive the Equivalent Speech Spectrum Level from
physical measurements.
%       The standard allows the use of alternative transforms,
such as the one illustrated above, where appropriate.
%
% Equivalent Noise Spectrum Level, N-prime
%       Similar to the Equivalent Speech Spectrum Level, the
Equivalent Noise Spectrum Level is the noise spectrum level at the
center of the listener's
%       head (when the listener is temporarily absent) that
produces an eardrum noise level equal to the eardrum noise level
actually present in the
%       listening situation to be analyzed. Sections 5.1, 5.2, and
5.3 give three examples of how to derive the Equivalent Speech
Spectrum
%       Level from physical measurements.
%
% Hannes Muesch, 2003 - 2005

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VERIFY INTEGRITY OF INPUT
VARIABLES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

[x,Nvar] = size(varargin);
CharCount = 0;
Ident = [];
for k = 1:Nvar
    if ischar(varargin{k}) & (length(varargin{k})==1)
        CharCount = CharCount + 1;
        Ident = [Ident; k];
    end
end
if Nvar/CharCount ~= 2
    error('Every input must be preceded by an identifying string')
else
    for n = 1:length(Ident)

```

```

        if upper(varargin{Ident(n)}) == 'N'      % Equivalent Noise
Spectrum Level (3.15)
            N = varargin{Ident(n)+1};
        elseif upper(varargin{Ident(n)}) == 'E' % Equivalent
Speech Spectrum Level (3.6)
            E = varargin{Ident(n)+1};
        elseif upper(varargin{Ident(n)}) == 'T' % Equivalent
Hearing Threshold Level [dBHL] (3.23)
            T = varargin{Ident(n)+1};
        elseif upper(varargin{Ident(n)}) == 'I' % Band Importance
function (3.1)
            I = varargin{Ident(n)+1};
        else
            error('Only ''E'', ''I'', ''N'', and ''T'' are valid
identifiers');
        end
    end
end

if isempty(who('E')), error('The Equivalent Speech Spectrum Level,
''E'', must be specified'); end
if isempty(who('N')), N = -50*ones(1,18);
end;
if isempty(who('T')), T = zeros(1,18);
end;
if isempty(who('I')), I = 1;
end;

N = N(:)'; T = T(:)'; E = E(:)';
if length(N) ~= 18, error('Equivalent Noise Spectrum Level: Vector
size incorrect'); end;
if length(T) ~= 18, error('Equivalent Hearing Threshold Level: Vector
size incorrect'); end;
if length(E) ~= 18, error('Equivalent Speech Spectrum Level: Vector
size incorrect'); end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% IMPLEMENTATION OF SPEECH
INTELLIGIBILITY INDEX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ===== THE NUMBERS IN PARENTHESIS REFER TO THE
SECTIONS IN THE ANSI STANDARD =====

% Define band center frequencies for 1/3rd octave procedure (Table 3)
f = [160 200 250 315 400 500 630 800 1000 1250 1600 2000, ...
     2500 3150 4000 5000 6300 8000];

% Define Internal Noise Spectrum Level (Table 3)
X = [0.6 -1.7 -3.9 -6.1 -8.2 -9.7 -10.8 -11.9 -12.5 -13.5 -15.4 -17.7,
     ...
     -21.2 -24.2 -25.9 -23.6 -15.8 -7.1];

% Self-Speech Masking Spectrum (4.3.2.1 Eq. 5)
V = E - 24;

% 4.3.2.2
B = max(V,N);

```

```

% Calculate slope parameter Ci (4.3.2.3 Eq. 7)
C = 0.6.*(B+10*log10(f)-6.353) - 80;

% Initialize Equivalent Masking Spectrum Level (4.3.2.4)
Z = [];
Z(1) = B(1);

% Calculate Equivalent Masking Spectrum Level (4.3.2.5 Eq. 9)
for i = 2:18
    Z(i) = 10*log10(10.^(0.1*N(i)) + ...
        sum(10.^(0.1*(B(1:(i-1))+3.32.*C(1:(i-1))))).*log10(0.89*f(i)./f(1:(i-1)))));
end

% Equivalent Internal Noise Spectrum Level (4.4 Eq. 10)
X = X + T;

% Disturbance Spectrum Level (4.5)
D = max(Z,X);

% Level Distortion Factor (4.6 Eq. 11)
L = 1 - (E - SpeechSptr('normal') - 10)./160;
L = min(1,L);

% 4.7.1 Eq. 12
K = (E-D+15)/30;
K = min(1,max(0,K));

% Band Audibility Function (7.7.2 Eq. 13)
A = L.*K;

% Speech Intelligibility Index (4.8 Eq. 14)
S = sum(BndImp(I).*A);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                     PRIVATE FUNCTIONS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function I = BndImp(tst)
% Band importance functions:
% tst = 1: Average speech as specified in Table 3
%       2: various nonsense syllable tests where most English
%          phonemes occur equally often
%       3: CID-22
%       4: NU6
%       5: Diagnostic Rhyme test
%       6: short passages of easy reading material
%       7: SPIN

if (nargin ~= 1)
    error('Incorrect # of input args to BndImp');
end

if ~(tst==1 | tst==2 | tst==3 | tst==4 | tst==5 | tst==6 | tst==7),
    error('Band Importance function must be integer between 1 and 7');
end

```

```

end

BIArr= [0.0083  0      0.0365  0.0168  0      0.0114  0
        0.0095  0      0.0279  0.013   0.024   0.0153  0.0255
        0.015   0.0153  0.0405  0.0211  0.033   0.0179  0.0256
        0.0289  0.0284  0.05    0.0344  0.039   0.0558  0.036
        0.044   0.0363  0.053   0.0517  0.0571  0.0898  0.0362
        0.0578  0.0422  0.0518  0.0737  0.0691  0.0944  0.0514
        0.0653  0.0509  0.0514  0.0658  0.0781  0.0709  0.0616
        0.0711  0.0584  0.0575  0.0644  0.0751  0.066   0.077
        0.0818  0.0667  0.0717  0.0664  0.0781  0.0628  0.0718
        0.0844  0.0774  0.0873  0.0802  0.0811  0.0672  0.0718
        0.0882  0.0893  0.0902  0.0987  0.0961  0.0747  0.1075
        0.0898  0.1104  0.0938  0.1171  0.0901  0.0755  0.0921
        0.0868  0.112   0.0928  0.0932  0.0781  0.082   0.1026
        0.0844  0.0981  0.0678  0.0783  0.0691  0.0808  0.0922
        0.0771  0.0867  0.0498  0.0562  0.048   0.0483  0.0719
        0.0527  0.0728  0.0312  0.0337  0.033   0.0453  0.0461
        0.0364  0.0551  0.0215  0.0177  0.027   0.0274  0.0306
        0.0185  0      0.0253  0.0176  0.024   0.0145  0];

I = BIArr(:,tst)';
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function E = SpeechSptr(VclEfirt)
% This function returns the standard speech spectrum level from Table
3

Ei=[32.41  33.81  35.29  30.77;
    34.48  33.92  37.76  36.65;
    34.75  38.98  41.55  42.5;
    33.98  38.57  43.78  46.51;
    34.59  39.11  43.3   47.4;
    34.27  40.15  44.85  49.24;
    32.06  38.78  45.55  51.21;
    28.3   36.37  44.05  51.44;
    25.01  33.86  42.16  51.31;
    23     31.89  40.53  49.63;
    20.15  28.58  37.7   47.65;
    17.32  25.32  34.39  44.32;
    13.18  22.35  30.98  40.8;
    11.55  20.15  28.21  38.13;
    9.33   16.78  25.41  34.41;
    5.31   11.47  18.35  28.24;
    2.59   7.67  13.87  23.45;
    1.13   5.07  11.39  20.72];

switch lower(VclEfirt)
case 'normal', E = Ei(:,1)';
case 'raised', E = Ei(:,2)';
case 'loud',   E = Ei(:,3)';
case 'shout',  E = Ei(:,4)';
otherwise, error('Identifier string to 'E' not recognized')
end

end

```

```
% EOF
```

## STOI

```
function d = STOI_1(s,n,sn,Fs)

if nargin<4,
    Fs=44100;
end

dbdif = ones(1,30);
dbdif = dbdif*(-abs(sn));

filtro_corrector = crea_filtro(dbdif,100,44100,0,44100); % creació
d'un filtre pla per fer l'atenuació del senyal
%hfvf =
fvtool(filtro_corrector,'FrequencyScale','log','color','white');

% Atenuació del senyal de veu o de soroll depenent del SNR
if sn > 0
    n = filter(filtro_corrector.Numerator,1,n);
elseif sn < 0
    s = filter(filtro_corrector.Numerator,1,s);
end

d = stoi(s,s+n,Fs);

end

%----- FUNCIÓ STOI -----
----
function d = stoi(x, y, fs_signal)
% d = stoi(x, y, fs_signal) returns the output of the short-time
% objective intelligibility (STOI) measure described in [1, 2],
where x
% and y denote the clean and processed speech, respectively, with
sample
% rate fs_signal in Hz. The output d is expected to have a monotonic
% relation with the subjective speech-intelligibility, where a
higher d
% denotes better intelligible speech. See [1, 2] for more details.
%
% References:
% [1] C.H.Taal, R.C.Hendriks, R.Heusdens, J.Jensen 'A Short-Time
% Objective Intelligibility Measure for Time-Frequency Weighted
Noisy
% Speech', ICASSP 2010, Texas, Dallas.
%
% [2] C.H.Taal, R.C.Hendriks, R.Heusdens, J.Jensen 'An Algorithm
for
% Intelligibility Prediction of Time-Frequency Weighted Noisy
Speech',
% IEEE Transactions on Audio, Speech, and Language Processing,
2011.
%
%
% Copyright 2009: Delft University of Technology, Signal & Information
```



```

% Processing Lab. The software is free for non-commercial use. This
program
% comes WITHOUT ANY WARRANTY.
%
%
%
% Updates:
% 2011-04-26 Using the more efficient 'taa_corr' instead of 'corr'

if length(x)~=length(y)
    error('x and y should have the same length');
end

% initialization
x          = x(:);           % clean speech column
vector
y          = y(:);           % processed speech
column vector

fs         = 10000;          % sample rate of
proposed intelligibility measure
N_frame    = 256;            % window support
K          = 512;            % FFT size
J          = 15;             % Number of 1/3 octave
bands
mn         = 150;            % Center frequency of
first 1/3 octave band in Hz.
H          = thirddoct(fs, K, J, mn); % Get 1/3 octave band
matrix
N          = 30;             % Number of frames for
intermediate intelligibility measure (Length analysis window)
Beta       = -15;            % lower SDR-bound
dyn_range  = 40;             % speech dynamic range

% resample signals if other samplerate is used than fs
if fs_signal ~= fs
    x      = resample(x, fs, fs_signal);
    y      = resample(y, fs, fs_signal);
end

% remove silent frames
[x y] = removeSilentFrames(x, y, dyn_range, N_frame, N_frame/2);

% apply 1/3 octave band TF-decomposition
x_hat    = stdft(x, N_frame, N_frame/2, K); % apply short-time DFT
to clean speech
y_hat    = stdft(y, N_frame, N_frame/2, K); % apply short-time DFT
to processed speech

x_hat    = x_hat(:, 1:(K/2+1)).'; % take clean single-
sided spectrum
y_hat    = y_hat(:, 1:(K/2+1)).'; % take processed
single-sided spectrum

X        = zeros(J, size(x_hat, 2)); % init memory for
clean speech 1/3 octave band TF-representation
Y        = zeros(J, size(y_hat, 2)); % init memory for
processed speech 1/3 octave band TF-representation

for i = 1:size(x_hat, 2)

```

```

X(:, i) = sqrt(H*abs(x_hat(:, i)).^2);          % apply 1/3 octave
bands as described in Eq.(1) [1]
Y(:, i) = sqrt(H*abs(y_hat(:, i)).^2);
end

% loop al segments of length N and obtain intermediate intelligibility
measure for all TF-regions
d_interm = zeros(J, length(N:size(X, 2)));
% init memory for intermediate intelligibility measure
c = 10^(-Beta/20);
% constant for clipping procedure

for m = N:size(X, 2)
    X_seg = X(:, (m-N+1):m);
    % region with length N of clean TF-units for all j
    Y_seg = Y(:, (m-N+1):m);
    % region with length N of processed TF-units for all j
    alpha = sqrt(sum(X_seg.^2, 2)./sum(Y_seg.^2, 2));
    % obtain scale factor for normalizing processed TF-region for all j
    aY_seg = Y_seg.*repmat(alpha, [1 N]);
    % obtain \alpha*Y_j(n) from Eq.(2) [1]
    for j = 1:J
        Y_prime = min(aY_seg(j, :), X_seg(j, :)+X_seg(j,
:)*c); % apply clipping from Eq.(3)
        d_interm(j, m-N+1) = taa_corr(X_seg(j, :).', Y_prime(:));
    % obtain correlation coeffecient from Eq.(4) [1]
    end
end

d = mean(d_interm(:));
% combine all intermediate intelligibility measures as in Eq.(4) [1]
end
%
function [A cf] = thirdoct(fs, N_fft, numBands, mn)
% [A CF] = THIRDOCT(FS, N_FFT, NUMBANDS, MN) returns 1/3 octave band
matrix
% inputs:
% FS:          samplerate
% N_FFT:       FFT size
% NUMBANDS:    number of bands
% MN:         center frequency of first 1/3 octave band
% outputs:
% A:          octave band matrix
% CF:         center frequencies

f = linspace(0, fs, N_fft+1);
f = f(1:(N_fft/2+1));
k = 0:(numBands-1);
cf = 2.^(k/3)*mn;
fl = sqrt((2.^(k/3)*mn).*2.^(k-1)/3)*mn;
fr = sqrt((2.^(k/3)*mn).*2.^(k+1)/3)*mn;
A = zeros(numBands, length(f));

for i = 1:(length(cf))
    [a b] = min((f-fl(i)).^2);
    fl(i) = f(b);
    fl_ii = b;

    [a b] = min((f-fr(i)).^2);
    fr(i) = f(b);

```

```

        fr_ii                = b;
        A(i,fl_ii:(fr_ii-1)) = 1;
end

rnk          = sum(A, 2);
numBands     = find((rnk(2:end)>=rnk(1:(end-1))) & (rnk(2:end)~=0)~=0,
1, 'last' )+1;
A            = A(1:numBands, :);
cf           = cf(1:numBands);

end

%
function x_stdft = stdft(x, N, K, N_fft)
%   X_STDFT = X_STDFT(X, N, K, N_FFT) returns the short-time
%   hanning-windowed dft of X with frame-size N, overlap K and DFT
size
%   N_FFT. The columns and rows of X_STDFT denote the frame-index and
%   dft-bin index, respectively.

frames       = 1:K:(length(x)-N);
x_stdft      = zeros(length(frames), N_fft);

w            = hanning(N);
x            = x(:);

for i = 1:length(frames)
    ii        = frames(i):(frames(i)+N-1);
    x_stdft(i, :) = fft(x(ii).*w, N_fft);
end

end

%
function [x_sil y_sil] = removeSilentFrames(x, y, range, N, K)
%   [X_SIL Y_SIL] = REMOVESILENTFRAMES(X, Y, RANGE, N, K) X and Y
%   are segmented with frame-length N and overlap K, where the maximum
energy
%   of all frames of X is determined, say X_MAX. X_SIL and Y_SIL are
the
%   reconstructed signals, excluding the frames, where the energy of a
frame
%   of X is smaller than X_MAX-RANGE

x            = x(:);
y            = y(:);

frames      = 1:K:(length(x)-N);
w           = hanning(N);
msk         = zeros(size(frames));

for j = 1:length(frames)
    jj        = frames(j):(frames(j)+N-1);
    msk(j)    = 20*log10(norm(x(jj).*w)./sqrt(N));
end

msk         = (msk-max(msk)+range)>0;
count       = 1;

x_sil       = zeros(size(x));
y_sil       = zeros(size(y));

```

```

for j = 1:length(frames)
    if msk(j)
        jj_i           = frames(j):(frames(j)+N-1);
        jj_o           = frames(count):(frames(count)+N-1);
        x_sil(jj_o)    = x_sil(jj_o) + x(jj_i).*w;
        y_sil(jj_o)    = y_sil(jj_o) + y(jj_i).*w;
        count          = count+1;
    end
end

x_sil = x_sil(1:jj_o(end));
y_sil = y_sil(1:jj_o(end));
end
%
function rho = taa_corr(x, y)
%   RHO = TAA_CORR(X, Y) Returns correlation coefficient between
column
%   vectors x and y. Gives same results as 'corr' from statistics
toolbox.
xn      = x-mean(x);
xn      = xn/sqrt(sum(xn.^2));
yn      = y-mean(y);
yn      = yn/sqrt(sum(yn.^2));
rho     = sum(xn.*yn);

end

```

## A.1.2 - Funcions secundaries

```

function dbdif =
calculo_ponderacion(dbpowsenyal,dbpowruido,tipo,difmin, factamp, amp, max
i)

if nargin<4,
    difmin=0;
end

if nargin<5,
    factamp=0.2;
end

dbdif = dbpowruido-dbpowsenyal; % càlcul de la diferencia de nivells
% de les senyals

switch tipo
    case 1
        for k=1:length(dbdif)
            if dbdif(k) > 0
                dbdif(k) = abs(dbdif(k))+difmin; % Aumenta el nivell
de la senyal fins el nivell del soroll
            elseif dbdif(k) < 0
                dbdif(k) = difmin; % Deixa la senyal al mateix nivell
            end
        end
    case 2
        if maxi == 1
            for k=1:length(dbdif)

```

```

                dbdif(k) = max(dbdif); % puja tota la senyal el nivell
de diferència màxima
            end
        else
            for k=1:length(dbdif)
                dbdif(k) = amp; % puja totes les bandes el nivell
indicat
            end
        end
    case 3
        for k=1:length(dbdif)
            dbdif(k) = dbpowruido(k)*factamp; % puja la senyal
depenent del espectre del soroll
        end

end

end

```

```
function dbpow = calculo_potencia(x,volsenyal)
```

```
[Hd3, Nfc] = filtro13octava();
```

```
for k=1:Nfc
    hq(k) = convert(Hd3(k), 'df1');
    y = filter(hq(k), x);
    power(k) = rms(y);
    dbpow(k) = 20*log10(power(k)) + volsenyal;
end

end

```

```
% ----- funcio filtro de tercio de octava -----
```

```
function [Hd3, Nfc] = filtro13octava()
```

```
f = fdesign.octave(3, 'Class 1', 'N,F0', 4, 1000, 44100);
```

```
F0 = validfrequencies(f);
```

```
Nfc = length(F0);
```

```
for i=1:Nfc,
    f.F0 = F0(i);
    Hd3(i) = design(f, 'butter');
end

```

```
end

```

```
function s_corregida = compara(s, potamp, liminf, limsup, deltaLp)
```

```
dbpowsenyal = calculo_potencia(s, potamp);
```

```
dbpowsenyal(isnan(dbpowsenyal)) = 0;
```

```
increment_find = Increment(deltaLp, liminf, limsup, dbpowsenyal);
```

```

filtro_corrector2 =
crea_filtro(increment_find,10000,44100,liminf,limsup);
s_corregida = filter(filtro_corrector2.Numerator,1,s);

end

function s_corregida = correccion( s, n, potamp, tipo, difmin ,
liminf, limsup, factamp, amp, maxi)

dbpowsenyal = calculo_potencia(s,potamp);
dbpowruido = calculo_potencia(n,potamp);

dbpowsenyal(isnan(dbpowsenyal)) = 0;
dbpowruido(isnan(dbpowruido)) = 0;

dbdif =
calculo_ponderacion(dbpowsenyal,dbpowruido,tipo,difmin,factamp, amp,
maxi);

filtro_corrector = crea_filtro(dbdif,5000,44100,liminf,limsup);

s_corregida = filter(filtro_corrector.Numerator,1,s);

end

function [ s, n ] = correccion_volumen( s, n, sn )

dbdif = ones(1,30);
dbdif = dbdif*(-abs(sn));

filtro_corrector = crea_filtro(dbdif,100,44100,0,44100);

if sn > 0
    n = filter(filtro_corrector.Numerator,1,n);
else
    s = filter(filtro_corrector.Numerator,1,s);
end

end

function filtre=crea_filtro(ponderacion,orden,fs,liminf,limsup)

    if nargin<2,
        orden=100;
    end

    if nargin<3,
        fs=44100;
    end

    if nargin<4,
        liminf=300;
    end
end

```

```

if nargin<5,
    limsup=3000;
end

[Hd3, Nfc, F0] = filtro13octava();

for i=1:length(ponderacion)
    if isnan(ponderacion(i))
        ponderacion(i) = 0;
    end
end

filtro_datos = [F0 ; round(ponderacion)];

for i=1:length(filtro_datos)
    if filtro_datos(1,i) < liminf || filtro_datos(1,i) > limsup
        filtro_datos(2,i) = 0;
    end
end

[f,m]=vectores_filtro(filtro_datos,fs);
filtre=dffir(dffir(firls(orden,f,m))); %creació del filtre

end

% CREACIÓ VECTORS FILTRE

function [f,m]=vectores_filtro(filtro_datos,fs)
% filtre_dades -> La primera fila seran les freqüències i la segona
% filera seran els nivells dB SPL de cadascuna de les freqüències

if nargin<2,
    fs=44100;
end

num_index=length(filtro_datos);

x=10*10^(1/10)*10^(1/20); %Inicialització x1 -
-> (f2 de tall d'una banda de terç d'octava centrada a 1
Hz). Per determinar aquests punts es fa servir base 10

f(1)=0;

for k=1:num_index,
    while x<filtro_datos(1,k),
        %càlcul de la f2 de tall per cada banda

        x=x*10^(1/10);
        %Anem saltant a la f2 de tall de la banda següent

    end
    for s=1:2,
        if k<=num_index-1,
            f(k*2+s-1)=x;
        end
        %f1 = f2 de la banda anterior
    end
end

```

```

        m((k)*2+s-2)=1/(10^(-filtro_datos(2,k)/20));
%càlcul dels factors multiplicatius per cada banda (invers de
la diferència del nivell de pressió sonora)
    end
end
f((num_index)*2)=fs/2;
f=f/(fs/2);
%Normalització del vectors de les freqüències entre 0 i 1 (
0 Hz fins a Nyquist)

end

function [Hd3, Nfc, F0] = filtro13octava()

f = fdesign.octave(3,'Class 1','N,F0',4,1000,44100);

F0 = validfrequencies(f);
Nfc = length(F0);

for i=1:Nfc,
    f.F0 = F0(i);
    Hd3(i) = design(f,'butter');
end

end

function [ s, n, potamp ] = igualacion_potencia( x, y , pot)

L = min(length(x),length(y));
s = x(1:L);
n = y(1:L);

power = rms(s);
powern = rms(n);

if powern < power
    s = s*(powern/power);
    dbpow = 20*log10(power);
    potamp = pot-dbpow; % potència necessaria per arribar al nivell de
potència assignat
else
    n = n*(power/powern);
    dbpown = 20*log10(powern);
    potamp = pot-dbpown; % potència necessaria per arribar al nivell
de potència assignat
end

end

function increment_find = Increment(DeltaLp, liminf, limsup,
dbpowsenyal)
[Hd3, Nfc, F0] = filtro13octava();

filtro_datos = [F0 ; ones(1,30)];
filtro_datos_no = [F0 ; ones(1,30)];

```



```

for i=1:length(filtro_datos)
    if filtro_datos(1,i) < liminf || filtro_datos(1,i) > limsup
        filtro_datos(2,i) = 0;
        %N = N-1;
    else
        filtro_datos_no(2,i) = 0;
    end
end

LpGlobal = 10*log10(sum(10.^(dbpowsenyal/10)));

LpiYES=filtro_datos(2,:).*dbpowsenyal;
LpiNO=filtro_datos_no(2,:).*dbpowsenyal;

error = .01;

for inc = 0:0.01:200
    if abs(10*log10(sum(10.^(LpiNO/10))+sum(10.^(LpiYES+inc)/10))-
(LpGlobal+DeltaLp)) < error
        inc_fi = inc;
    end
end

increment_find = inc_fi*filtro_datos(2,:);

end

function scor = normalizacion(s)

scor = (s - mean(s)) / max(abs((s - mean(s)))));

end

```

## ANNEX A.2 – Codi GUI

```

function varargout = guidel(varargin)
% GUIDEL MATLAB code for guidel.fig
%   GUIDEL, by itself, creates a new GUIDE1 or raises the existing
%   singleton*.
%
%   H = GUIDEL returns the handle to a new GUIDE1 or the handle to
%   the existing singleton*.
%
%   GUIDEL('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in GUIDEL.M with the given input
arguments.
%
%   GUIDEL('Property','Value',...) creates a new GUIDE1 or raises
the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before guidel_OpeningFcn gets called. An

```

```

% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to guidel_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 guidel

% Last Modified by GUIDE v2.5 05-Jun-2015 09:51:26

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

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

% Declaración de las variables

handles.variables.audio=[];
handles.variables.r=[];
handles.variables.fs_original=[];

handles.variables.audio_sinmejora=[];
handles.variables.audio_conmejora1=[];
handles.variables.audio_conmejora2=[];
handles.variables.audio_conmejora3=[];

%centramos la ventana:
%centrar_figura(hObject);

```

```

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes guidata wait for user response (see UIRESUME)
% uiwait(handles.figure1);

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

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

[nombre_archivo,ruta]=uigetfile({'*.wav'},'Abrir archivo de
audio','/Users/carlosvegacarrera/Desktop/PFG/Ruido');
if isequal(nombre_archivo,0)
    return
else
    audiocompleto = strcat (ruta, nombre_archivo);
    handles.variables.r = wavread(audiocompleto);
    set(handles.cargadoR,'visible','on');
end

guidata(hObject, handles);

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

[nombre_archivo,ruta]=uigetfile({'*.wav'},'Abrir archivo de
audio','/Users/carlosvegacarrera/Desktop/PFG/Senyal');
if isequal(nombre_archivo,0)
    return
else
    audiocompleto = strcat (ruta, nombre_archivo);
    [handles.variables.audio, handles.variables.fs_original] =
wavread(audiocompleto);
    set(handles.cargadoS,'visible','on');
end

guidata(hObject, handles);

function SNRdB_Callback(hObject, eventdata, handles)
% hObject handle to SNRdB (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 SNRdB as text
% str2double(get(hObject,'String')) returns contents of SNRdB
as a double
res = str2num(char(get(hObject,'String')));
res = 10^(res/10);
set(handles.SNR,'String',num2str(res));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function SNRdB_CreateFcn(hObject, eventdata, handles)
% hObject handle to SNRdB (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 SNR_Callback(hObject, eventdata, handles)
% hObject handle to SNR (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 SNR as text
% str2double(get(hObject,'String')) returns contents of SNR as
a double

res = str2num(char(get(hObject,'String')));
res = 10*log10(res);
set(handles.SNRdB,'String',num2str(res));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function SNR_CreateFcn(hObject, eventdata, handles)
% hObject handle to SNR (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 during object creation, after setting all properties.
function uipanel5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes when selected object is changed in uipanel5.
function uipanel5_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uipanel5
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if
none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

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

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

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

```

```

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

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

% --- Executes during object creation, after setting all properties.
function resultado_mejoral_CreateFcn(hObject, eventdata, handles)
% hObject      handle to resultado_mejoral (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 liminf_Callback(hObject, eventdata, handles)
% hObject      handle to liminf (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 liminf as text
%           str2double(get(hObject,'String')) returns contents of liminf
as a double

% --- Executes during object creation, after setting all properties.
function liminf_CreateFcn(hObject, eventdata, handles)
% hObject      handle to liminf (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 limsup_Callback(hObject, eventdata, handles)
% hObject    handle to limsup (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 limsup as text
%         str2double(get(hObject,'String')) returns contents of limsup
as a double

% --- Executes during object creation, after setting all properties.
function limsup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to limsup (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 play1.
function play1_Callback(hObject, eventdata, handles)
% hObject    handle to play1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

audio = handles.variables.audio_sinmejora;
audio = normalizacion(audio);
Fs = handles.variables.fs_original;
if isempty(audio)
    errordlg('No se ha calculado la señal','Error 1');
else
    sound(audio,Fs);
end

% --- Executes on button press in play2.
function play2_Callback(hObject, eventdata, handles)
% hObject    handle to play2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

audio = handles.variables.audio_conmejoral;
audio = normalizacion(audio);
Fs = handles.variables.fs_original;
if isempty(audio)
    errordlg('No se ha calculado la señal','Error 2');
else
    sound(audio,Fs);
end

% --- Executes on button press in play3.
function play3_Callback(hObject, eventdata, handles)
% hObject    handle to play3 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

audio = handles.variables.audio_conmejora2;
audio = normalizacion(audio);
Fs = handles.variables.fs_original;
if isempty(audio)
    errordlg('No se ha calculado la señal','Error 2');
else
    sound(audio,Fs);
end

% --- Executes on button press in play4.
function play4_Callback(hObject, eventdata, handles)
% hObject handle to play4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
audio = handles.variables.audio_conmejora3;
audio = normalizacion(audio);
Fs = handles.variables.fs_original;
if isempty(audio)
    errordlg('No se ha calculado la señal','Error 2');
else
    sound(audio,Fs);
end

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

% --- Executes during object creation, after setting all properties.
function nivelse_CreateFcn(hObject, eventdata, handles)
% hObject handle to nivelse (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 difmin_Callback(hObject, eventdata, handles)
% hObject handle to difmin (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 difmin as text

```



```

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

% --- Executes during object creation, after setting all properties.
function difmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to difmin (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 calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

sn = str2num(char(get(handles.SNRdB,'String')));
pot = str2num(char(get(handles.nivelse,'String')));

if isempty(sn)
    errordlg('Rellena la relación señal-ruido','Error 1');
else
    if isempty(handles.variables.audio) ||
isempty(handles.variables.r) % Si no se ha leído alguno de los 2
        errordlg('Carga el audio y el ruido','Error 2');
    else
        [ s, n, potamp ] = igualacion_potencia(
handles.variables.audio, handles.variables.r , pot);

        [ s, n ] = correccion_volumen( s, n, sn );

        switch get(get(handles.uipanel5,'SelectedObject'),'Tag')
            case 'AI'
                display('Articulation Index');
                res = AI(s , n , sn , potamp);
                set(handles.resultado,'String',num2str(res));

            case 'SII'
                display('SII');
                res = SII_1(s , n , sn, potamp);
                set(handles.resultado,'String',num2str(res));

            case 'STOI'
                display('STOI');
                res = STOI_1(s , n , sn );
                set(handles.resultado,'String',num2str(res));
        end
        handles.variables.audio_sinmejora = s+n;
    end
end
end

```

```

guidata(hObject, handles);

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

sn = str2num(char(get(handles.SNRdB, 'String')));
pot = str2num(char(get(handles.nivelse, 'String')));

liminf = str2num(char(get(handles.liminf, 'String')));
limsup = str2num(char(get(handles.limsup, 'String')));

difmin = str2num(char(get(handles.difmin, 'String')));

switch get(get(handles.tipomejora, 'SelectedObject'), 'Tag')
    case 'mejora1'
        tipo = 1;
    case 'mejora2'
        tipo = 2;
    case 'mejora3'
        tipo = 3;
end

maxi = get(handles.checkbox1, 'Value');

if isempty(sn)
    errordlg('Rellena la relación de señal/ruido', 'Error 1');
else
    if isempty(handles.variables.audio) ||
        isempty(handles.variables.r) % Si no se ha leído alguno de los 2
        errordlg('Carga el audio y el ruido', 'Error 2');
    else
        [ s, n, potamp ] = igualacion_potencia(
handles.variables.audio, handles.variables.r , pot);

        [ s, n ] = correccion_volumen( s, n, sn );

        factamp =
str2num(char(get(handles.factamp, 'String')));
        amp =
str2num(char(get(handles.amplificacion2, 'String')));

        if factamp > 1 || factamp < 0
            errordlg('El factor de amplificación debe ser un
numero entre 0 y 1', 'Error 3');
        else
            s_corregida = correccion( s, n, potamp, tipo,
difmin , liminf, limsup, factamp, amp, maxi);

            A = 20*log10(rms(s));
            B = 20*log10(rms(s_corregida));

            set(handles.amplificacion, 'String', num2str(abs(B-
A)));
        end
    end
end

```

```

switch get(get(handles.uipanel5,'SelectedObject'),'Tag')
    case 'AI'
        display('Articulation Index');
        res = AI(s_corregida , n , sn , potamp);

        case 'SII'
            display('SII');
            res = SII_1(s_corregida , n , sn , potamp);

        case 'STOI'
            display('STOI');
            res = STOI_1(s_corregida , n , sn );

end

handles.variables.audio_sinmejora = s+n;

if tipo == 1
    set(handles.resultado_mejora1,'String',num2str(res));
    handles.variables.audio_conmejora1 = s_corregida+n;
elseif tipo == 2
    set(handles.resultado_mejora2,'String',num2str(res));
    handles.variables.audio_conmejora2 = s_corregida+n;
else
    set(handles.resultado_mejora3,'String',num2str(res));
    handles.variables.audio_conmejora3 = s_corregida+n;
end

end

end

guidata(hObject, handles);

function resultado_mejora2_Callback(hObject, eventdata, handles)
% hObject    handle to resultado_mejora2 (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 resultado_mejora2
as text
%         str2double(get(hObject,'String')) returns contents of
resultado_mejora2 as a double

% --- Executes during object creation, after setting all properties.
function resultado_mejora2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to resultado_mejora2 (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 compara.
function compara_Callback(hObject, eventdata, handles)
% hObject handle to compara (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
sn = str2num(char(get(handles.SNRdB,'String')));
pot = str2num(char(get(handles.nivelse,'String')));

liminf = str2num(char(get(handles.liminf,'String')));
limsup = str2num(char(get(handles.limsup,'String')));

difmin = str2num(char(get(handles.difmin,'String')));

switch get(get(handles.uipanel10,'SelectedObject'),'Tag')
    case 'comparal'
        comp = 1;
    case 'compara3'
        comp = 3;
end

maxi = get(handles.checkbox1, 'Value');

if isempty(sn)
    errordlg('Rellena la relación de señal/ruido','Error 1');
else
    if isempty(handles.variables.audio) ||
isempty(handles.variables.r) % Si no s'ha llegit algun dels 2
        errordlg('Carga el audio y el ruido','Error 2');
    else
        [ s, n, potamp ] = igualacion_potencia(
handles.variables.audio, handles.variables.r , pot);

        [ s, n ] = correccion_volumen( s, n, sn );

        factamp =
str2num(char(get(handles.factamp,'String')));
        amp =
str2num(char(get(handles.amplificacion2,'String')));

        if factamp > 1 || factamp < 0
            errordlg('El factor de amplificación debe ser un
numero entre 0 y 1','Error 3');
        else
            s_corregida = correccion( s, n, potamp, comp,
difmin , liminf, limsup, factamp, amp, maxi);

            A = 20*log10(rms(s));
            B = 20*log10(rms(s_corregida));

            deltaLp = abs(B-A);

set(handles.amplificacion,'String',num2str(deltaLp));

```

```

, deltaIp);

s_corregida2 = compara( s, potamp, liminf, limsup

dbpow = calculo_potencia(s_corregida2,potamp);
dbpow(isnan(dbpow)) = 0;

C = 20*log10(rms(s_corregida2));
display(B-C);
end

switch
get(get(handles.uipanel5, 'SelectedObject'), 'Tag')
case 'AI'
display('Articulation Index');
res1 = AI(s_corregida , n , sn , potamp);
res2 = AI(s_corregida2 , n , sn , potamp);

case 'SII'
display('SII');
res1 = SII_1(s_corregida , n , sn , potamp);
res2 = SII_1(s_corregida2 , n , sn , potamp);

case 'STOI'
display('STOI');
res1 = STOI_1(s_corregida , n , sn );
res2 = STOI_1(s_corregida2 , n , sn );
end

handles.variables.audio_sinmejora = s+n;

handles.variables.audio_conmejora2 = s_corregida2+n;

display(comp);

if comp == 1

set(handles.resultado_mejora1, 'String', num2str(res1));
set(handles.resultado_mejora2, 'String', num2str(res2));
handles.variables.audio_conmejora1 =
s_corregida+n;
else

set(handles.resultado_mejora3, 'String', num2str(res1));
set(handles.resultado_mejora2, 'String', num2str(res2));
handles.variables.audio_conmejora3 =
s_corregida+n;
end
end
end

guidata(hObject, handles);

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

% --- Executes during object creation, after setting all properties.
function amplificacion_CreateFcn(hObject, eventdata, handles)
% hObject handle to amplificacion (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 resultado_mejora3_Callback(hObject, eventdata, handles)
% hObject handle to resultado_mejora3 (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 resultado_mejora3
as text
% str2double(get(hObject,'String')) returns contents of
resultado_mejora3 as a double

% --- Executes during object creation, after setting all properties.
function resultado_mejora3_CreateFcn(hObject, eventdata, handles)
% hObject handle to resultado_mejora3 (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 factamp_Callback(hObject, eventdata, handles)
% hObject handle to factamp (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 factamp as text

```

```

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

% --- Executes during object creation, after setting all properties.
function factamp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to factamp (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 when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

selection = questdlg('El programa se esta cerrando. Desea
continuar?','Calculador de indices','Sí','No','Sí');

switch selection,
    case 'Sí',
        delete(hObject);
    case 'No',
        return
end

function amplificacion2_Callback(hObject, eventdata, handles)
% hObject    handle to amplificacion2 (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 amplificacion2 as
text
%         str2double(get(hObject,'String')) returns contents of
amplificacion2 as a double

% --- Executes during object creation, after setting all properties.
function amplificacion2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplificacion2 (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

```

```
end
```

```
% --- Executes on button press in checkbox1.  
function checkbox1_Callback(hObject, eventdata, handles)  
% hObject    handle to checkbox1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hint: get(hObject,'Value') returns toggle state of checkbox1
```