

# Pengkodean Sinyal Suara Dengan Standar MPEG-4 Audio HVXC

Made Santo Gitakarma, NIM 13298050, Teknik Telekomunikasi  
Jurusan Teknik Elektro – ITB

## ABSTRAK

Seiring perkembangan teknologi digital untuk aplikasi-aplikasi multimedia, berbagai pengkodean sinyal suara bermunculan. Namun kebanyakan pengkodean suara yang ada hanya mendukung fungsi “kompresi” tunggal. MPEG-4 adalah standar ISO/IEC yang dibuat oleh MPEG (Moving Picture Experts Group) untuk aplikasi multimedia yang meliputi video dan audio (suara dan musik). MPEG-4 memiliki algoritma pengkodean suara dengan efisiensi pengkodean tinggi dan fungsional yang banyak (multiple) yang berperan penting untuk efisiensi penggunaan bandwidth dan memunculkan aplikasi-aplikasi baru multimedia. Standar MPEG-4 Audio dengan metode pengkodean parametrik audio (untuk aplikasi very low bit rate) dibagi menjadi dua bagian yaitu HVXC (Harmonic Vector eXcitation Coding) untuk pengkodean sinyal suara dan HILN (Harmonic and Individual Line plus Noise) untuk pengkodean sinyal musik. Pada tulisan ini akan dibahas proses pada encoder dan decoder pengkodean sinyal suara menggunakan HVXC. Sinyal input yang digunakan memiliki bandwidth Narrowband (300-3400Hz) yang dinyatakan secara digital ke bentuk frame-frame pada frekuensi sampling 8 kHz dan bit rate tetap 2.0 kbps atau 4.0kbps.

Keyword : encoder, decoder, vektor harmonik ..

## A. PENDAHULUAN

Hadirnya kompresi audio disebabkan data audio digital kualitas tinggi memerlukan ruang hard disk yang besar untuk menyimpan (atau bandwidth channel yang besar untuk transmisi) [12]. Perhatikan contoh singkat berikut. Anda ingin menyampling lagu favorit 1menit dan menyimpannya di harddisk. Karena ingin kualitas CD, Anda sampling pada 44.1 kHz, stereo, dengan 16 bits per sampel. 44.100 Hz berarti Anda punya 44.100 nilai per detik didatangkan dari sound card Anda (atau file input). Kalikan dengan dua karena Anda punya dua channel. Kalikan lagi dengan dua karena Anda punya dua byte per nilai (arti dari 16 bit). Lagu tersebut akan menghabiskan:

$$44.100 \text{ samples/s} * 2 \text{ channels} * 2 \text{ bytes/sample} \\ * 60 \text{ s/min} = \text{sekitar } 10 \text{ MBytes}$$

ruang tempat menyimpan pada harddisk. Dan jika Anda ingin mendownloadnya melalui internet, misalkan dengan modem 28.8kbps, maka waktu yang dibutuhkan:

$$10.000.000\text{bytes} * 8 \text{ bits/byte} / (28.800 \text{ bits/s}) \\ * 60 \text{ s/min} = \text{sekitar } 49 \text{ menit.}$$

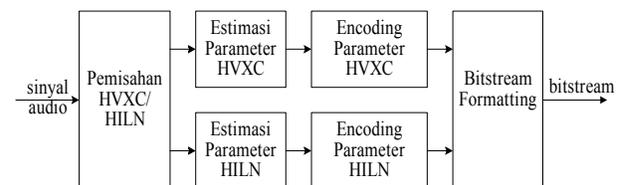
### Hanya untuk men-download 1 menit musik stereo!

Kini telah dikenal mp3 (MPEG 1 Layer 3) [11] yang memiliki faktor kompresi 12 (dari kualitas CD) dengan bit rate 128 kbps. Namun dengan coder parametrik didapat efisiensi pengkodean sangat tinggi dengan faktor kompresi 64 pada bit rate 2 kbps. Memang kualitas suaranya tidak sebaik mp3, namun lebih baik daripada CELP (FS-1016) pada 4.8 kbps. Efisiensi pengkodean yang tinggi terutama diperlukan untuk aplikasi multimedia dengan komunikasi real time seperti Voice Mail, Internet Telephony, komunikasi Mobile dan Satelit [4][7].

HVXC adalah bagian standar MPEG-4 Audio [2] yang digunakan untuk pengkodean suara Narrowband dengan frekuensi sampling 8 kHz pada bit rate 2.0 atau 4.0 kbps. Coder HVXC berdasarkan pada pengkodean prediksi linier, namun menggunakan pendekatan yang berbeda untuk frame-frame voiced dan unvoiced. Komponen eksitasi unvoiced dinyatakan dalam vektor-vektor stochastic codebook sama halnya dengan teknik pengkodean konvensional CELP. Komponen voiced dikodekan dalam bentuk spectral envelope sinyal eksitasi.

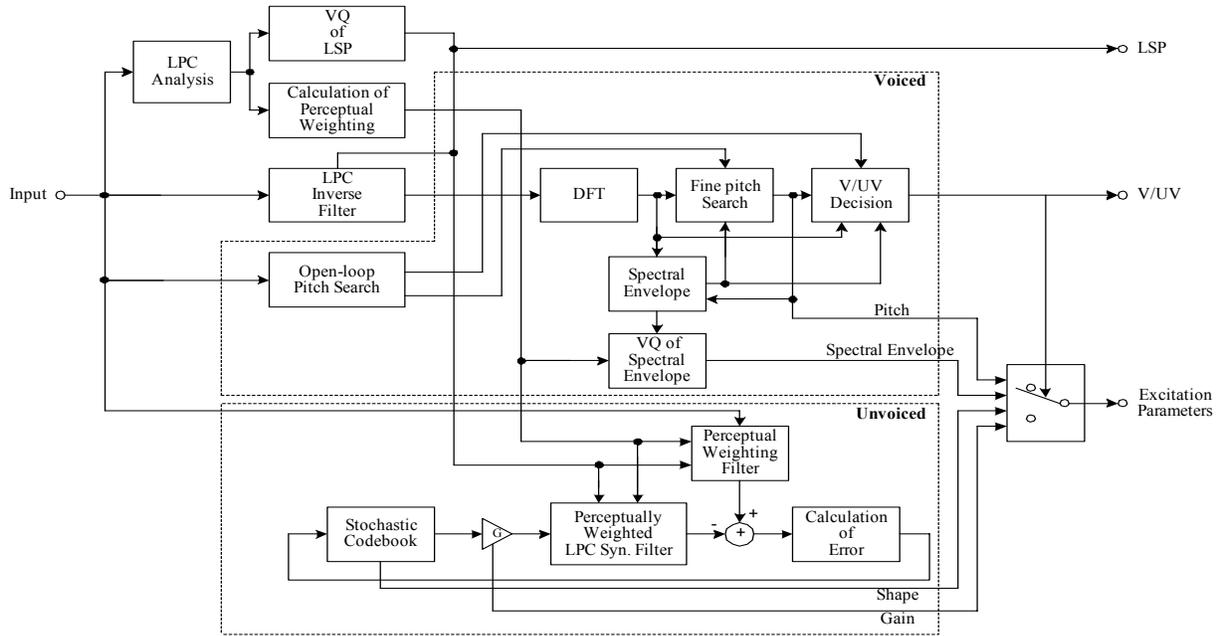
## B. ENCODER HVXC

Sebelum memasuki encoder HVXC, pertama kali sinyal input audio dipisah menjadi dua bagian di dalam encoder *parametrik* (lihat gambar 1) [3] yang dikode dengan tool-tool HVXC dan HILN. Format bitstream (aliran bit) yang dihasilkan bisa dioperasikan pada hanya HVXC saja, hanya HILN saja, atau mode kombinasi keduanya.



Gambar 1 – Diagram blok umum encoder parametrik

Pada encoder HVXC (gambar 2), sinyal disampling dengan frekuensi sampling 8 kHz dibentuk menjadi frame-frame dengan panjang 256 sampel dan interval 160 sampel (lihat gambar 3).



Gambar 2 – Diagram blok Encoder HVXC

### B1. Normalisasi

Frame-frame kemudian dinormalisasi yang meliputi tiga tahap operasi antara lain analisis Linear Predictive Coding (LPC), kuantisasi parameter Line Spectral Pair (LSP), dan filter inverse LPC. Di dalam LPC Analysis, data sinyal input diwindow per frame menggunakan window Hamming:

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(2\pi \frac{n}{N-1}\right) & ; 0 \leq n \leq N-1 \\ 0 & ; \text{lainnya} \end{cases} \quad (1)$$

Koefisien LPC orde 10 dihitung untuk setiap frame dari sinyal input yang sudah diwindow dengan metode *autokorelasi*. Koefisien-koefisien LPC ini diubah menjadi parameter-parameter LSP. Parameter LSP ini kemudian masing-masing dikuantisasi dengan Vector Quantization (VQ). Filter sintesis LPC:

$$H(z) = \frac{1}{\sum_{n=0}^P \alpha_n z^{-n}} \quad (2)$$

dimana  $\alpha_n$  adalah koefisien prediksi linier dan  $P$  adalah orde koefisien (=10).

Pada layer base, ada dua metode kuantisasi LSP yaitu *two-stage VQ tanpa prediksi interframe*, dan *two-stage VQ dengan prediksi interframe*. Dalam proses encoding, kedua metode dicoba untuk mengkuantisasi parameter LSP dan ditentukan metode mana yang harus dipakai dengan membandingkan error kuantisasi. Error kuantisasi dihitung sebagai jarak euclidean terbebani. Pada layer enhancement, vektor kuantizer dimensi 10, yang

memiliki codebook 8 bit ditambahkan di bawah skema kuantizer current LSP dari coder 2 kbps.

Parameter-parameter LSP kemudian diubah menjadi parameter alfa ( $\alpha$ ) untuk membentuk filter inverse LPC. Sinyal residu LPC (256 titik per frame) dihitung dengan mem-filter inverse sinyal input kemudian diwindow Hamming untuk menghitung spektrum daya. Fungsi transfer filter inverse adalah:

$$A(z) = \sum_{n=0}^P \alpha_n z^{-n} \quad (3)$$

### B2. Estimasi Pitch

Untuk mendapatkan estimasi pertama nilai pitch lag, nilai autokorelasi dari sinyal residu LPC dihitung. Dengan nilai lag ini, *open loop pitch* diestimasi. Tracking pitch dihasilkan dalam proses perkiraan / estimasi pitch sehingga pitch-nya lebih reliable.

### B3. Ekstraksi Magnituda Harmonik

Ekstraksi magnituda harmonik terdiri dari dua tahap yaitu *fine pitch search* dan *estimasi spectral envelope*. Pada tahap *fine pitch search*, nilai pitch fraksional diestimasi menggunakan *open loop integer pitch lag*. Ukuran step fraksi adalah 0.25. Ini didapat dari meminimalisasi error antara spektrum hasil sintesis dengan spektrum original. Disini nilai pitch dan magnituda spectral diestimasi secara simultan. Pada tahap estimasi spectral envelope, sinyal residu LPC ditransformasi dengan DFT 256 point untuk mendapatkan spektrum original. Dengan menggunakan spektrum original, estimasi spectral envelope didapat pada bagian fine pitch search. Spectral envelope adalah set magnituda spektral hasil estimasi pada tiap harmonik.

#### B4. Pembobotan Perceptual

Respon frekuensi filter pembobotan perceptual (perceptual weighting) dihitung untuk penggunaan vektor kuantisasi terbebani dalam harmonic spectral envelope. Fungsi transfer filter pembobotan perceptual adalah sebagai berikut:

$$w(z) = \frac{\sum_{n=0}^P \alpha_n A^n z^{-n}}{\sum_{n=0}^P \alpha_n B^n z^{-n}} \quad (4)$$

dimana  $A=0.9$  dan  $B=0.4$ . Pada tool ini, respon frekuensi  $w(z)h(z)$  dihitung dan dikeluarkan sebagai array komponen matriks  $WH$ .

#### B5. Encoder VQ Harmonik

Proses encoding kuantisasi vektor (VQ) harmonik terdiri dari dua tahap yaitu konversi dimensi dan kuantisasi vektor-vektor residu. Jumlah titik yang menyusun spectral envelope bervariasi tergantung nilai pitch karena spectral envelope merupakan set estimasi magnituda pada tiap harmonik dengan jumlah harmonik berkisar antara 8 sampai 60. Untuk mengkuantisasi vektor spectral envelope, coder harus mengubahnya ke jumlah konstan untuk dimensi VQ tetap. Interpolasi band terbatas digunakan untuk konversi frekuensi sampling dan mendapatkan dimensi tetap vektor-vektor spectral. Jumlah titik yang menyatakan shape spectral envelope harus dimodifikasi tanpa mengubah shape. Untuk itu konverter dimensi untuk spectral envelope dengan kombinasi LPF dan interpolator linier orde 1 digunakan. FIR LPF dengan 7 set koefisien yang masing-masing berisi 8 koefisien digunakan untuk stage pertama 8 kali over-sampling. Tujuh set koefisien filter didapat dari pengelompokan setiap 8 koefisien dari window sinc,  $coef[i]$ , dengan offset 1 sampai 7, dimana

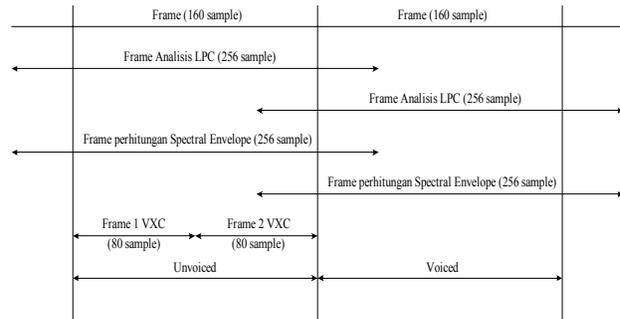
$$coef[i] = \frac{\sin \pi(i-32)/8}{\pi(i-32)/8} (0.5 - 0.5 \cos 2\pi i / 64) \quad 0 \leq i \leq 64 \quad (5)$$

#### B6. Keputusan V/UV

Keputusan V/UV dibuat per 20ms frame berdasarkan kesamaan shape spektrum sintesis dan spektrum original, daya sinyal, maksimum autokorelasi sinyal residu LPC yang dinormalisasi dengan daya sinyal residu, dan jumlah zero crossing.

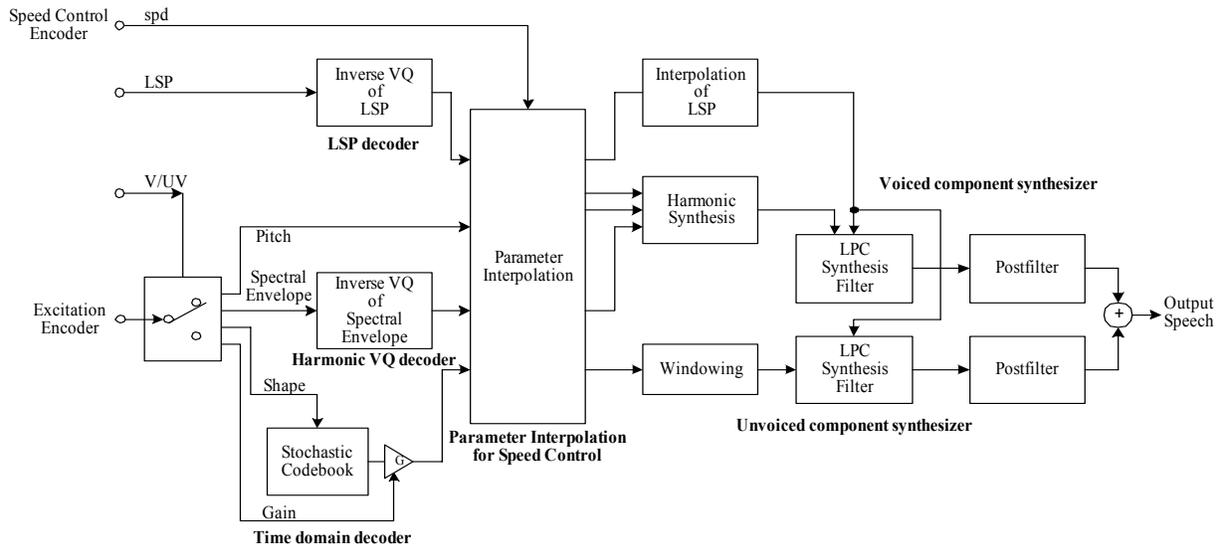
#### C. DECODER HVXC

Decoder HVXC (lihat gambar 4) mendukung mode *normal/low delay* untuk encode dan decode, kombinasi untuk salah satu mode delay pada 2.0-4.0 kbps dengan skema scalable. Struktur frame setiap mode delay dapat dilihat pada gambar 3.



Gambar 3 - Struktur frame HVXC

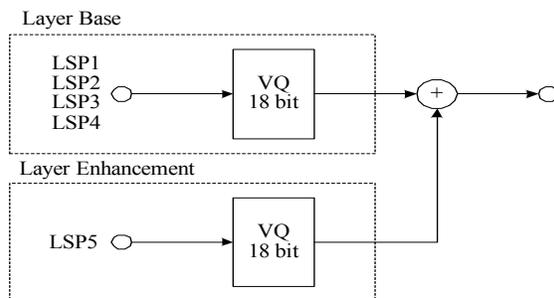
Untuk mendapatkan parameter-parameter LSP, struktur quantizer multistage digunakan dan vektor-vektor



Gambar 4 – Diagram blok Decoder HVXC

output dari setiap stage harus dijumlahkan. Apabila bit rate-nya 2 kbps, LSP dari *frame sekarang*, yang dikode dengan split dan vektor kuantisasi two-stage, didecode dengan proses pendecodean two-stage. Untuk kasus 4 kbps, vektor kuantiser 10-dimensi, yang memiliki codebook 8 bit, ditambahkan di bawah skema kuantiser LSP dari coder 2.0 kbps. Bit rate LSP dinaikkan dari 18bits/20msec hingga 26bits/20msec. Parameter-parameter LSP kemudian diinterpolasi secara linier yang diupdate setiap 2.5 ms untuk segmen voiced. Interpolated LSP dikonversi ke bentuk langsung koefisien prediksi linier  $\alpha_n$ .

### C1. Decoder LSP



Gambar 5 – Decoder LSP

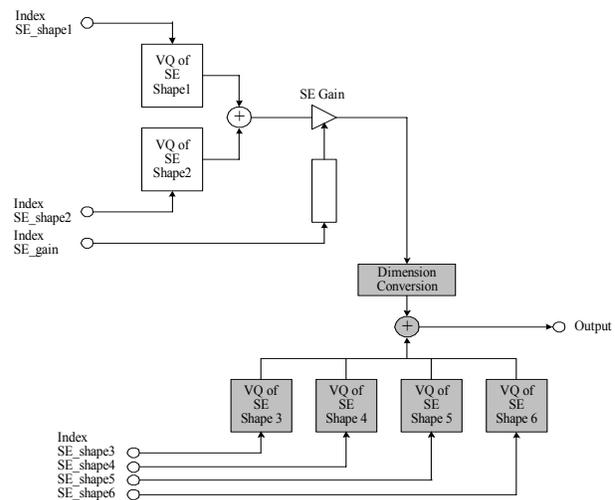
Proses pendecodean parameter-parameter LSP pada layer base untuk bit rate 2.0 kbps dan tambahan layer enhancement untuk bit rate 4.0 kbps dapat dilihat pada gambar 5. Prosesnya adalah sebagai berikut. LSP dari *frame sekarang*, yang dikode dengan split dan vektor kuantisasi two-stage, didecode dengan proses pendecodean two-stage. Pada stage pertama, vektor LSP di-decode dengan melihat tabel. Pada stage kedua, ada dua tipe proses pendecodean, yaitu proses pendecodean VQ tanpa prediksi interframe dan VQ dengan prediksi interframe. Proses pendecodean VQ tanpa prediksi interframe yaitu untuk mendapatkan LSP dari *frame sekarang*, vektor hasil decode pada stage kedua ditambahkan vektor LSP stage pertama hasil decode. Proses pendecodean VQ dengan prediksi interframe yaitu untuk mendapatkan LSP dari *frame sekarang*, vektor hasil decode dari stage kedua ditambahkan pada vektor LSP. Vektor ini diprediksi dari LSP hasil decode pada *frame sebelum* dan vektor stage pertama hasil decode.

LSP hasil decode distabilisasi untuk memastikan stabilitas filter sintesis LPC yang diperoleh dari LSP hasil decode. LSP hasil decode disusun meningkat, mempunyai jarak minimum antara koefisien yang berpasangan.

Setelah proses pendecodean, LSP hasil decode harus disimpan dalam memori, karena mereka berguna untuk prediksi pada frame berikutnya.

### C2. Decoder VQ Harmonik

Proses pendecodean pada decoder VQ harmonik untuk kasus 2 kbps, yaitu *kuantisasi vektor inverse dari vektor-vektor residu dan konversi dimensi*. Pada kasus 4 kbps, *kuantizer inverse tambahan* digunakan. Operasi dari masing-masing tahap dijelaskan pada gambar 6.



Gambar 6 - Vector Quantization dari Spectral Envelope

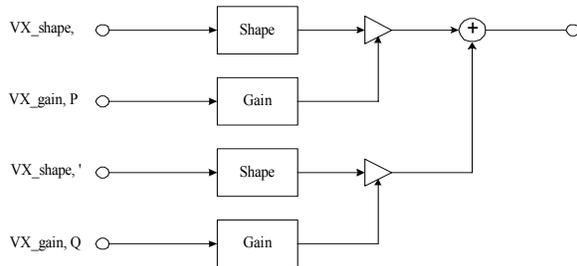
Untuk kuantisasi magnituda spektral harmonik, decoder 2.0 kbps menggunakan kombinasi *two-stage shape vector quantizer* dan *scalar gain quantizer*, dimana setiap bentuk (shape) codebook adalah 4 bits dan gain codebook. Dimensi shape codebook adalah tetap (=44). Inverse vector quantization pertama kali dibawa keluar (carried out) untuk mendapatkan *fixed dimension spectral vector*. Saat bit rate-nya 2 kbps, two-stage VQ pertama dan gain quantizer digunakan (SE\_shap1, SE\_shape2, dan SE\_gain). Dua shape vektor ditambahkan lalu dikalikan dengan gain. Untuk mendapatkan spectral vector dari dimensi original, konversi dimensi kemudian dipergunakan pada *fixed dimension spectral vector*. Untuk mode 4.0 kbps, dekuantisasi magnituda harmonik dengan dimensi tetap (=44) pertama kali dikonversi ke dimensi vektor harmonik original, yang bervariasi *frame by frame*. Output stage tambahan dengan skema split VQ dikomposisi dari empat vector quantizer (SE\_Shape3, SE\_Shape4, SE\_Shape5, SE\_Shape6) ditambahkan ke output *dimension converted quantizer* dari skema 2.0 kbps.

Untuk mendapatkan jumlah variabel magnituda harmonik, decoder harus mengubah *fixed-dimension codevector* menjadi *variable dimension vector*. Jumlah point yang menyatakan shape dari spectral envelope, harus dimodifikasi tanpa mengubah shape. Untuk itu, *dimension converter* untuk spectral envelope dengan kombinasi low pass filter dan interpolator linier orde 1 digunakan.

Filter FIR mengurangi perhitungan, yaitu hanya point-point yang digunakan pada stage berikutnya dihitung. Point-point tersebut berdekatan kiri dan kanan dari output terakhir pada *dimension converter*. Pada stage over-sampling kedua, interpolasi linier orde 1 dipergunakan untuk mendapatkan point-point output yang penting. Dalam hal ini, vektor-vektor spectral dimensi-variabel dari vektor-vektor spectral dimensi-tetap (= 44) didapat. Modifikasi pitch bisa dilakukan hanya dengan memodifikasi frekuensi fundamental target  $w_0$  dan jumlah harmonik yang bersesuaian menurut faktor modifikasi pitch  $pch\_mod$  sebelum konversi dimensi. Frekuensi fundamental target yang dimodifikasi untuk perubahan pitch kemudian dihitung:  $w_0 * pch\_mod$ .

### C3. Decoder Time Domain

Untuk segmen unvoiced pada suara, skemanya menggunakan prinsip VXC (Vector Excitation Coding). Decoder time domain menghasilkan waveform eksitasi untuk porsi unvoiced dengan melihat (look up) tabel menggunakan indeks yang ditransmisikan. Vektor shape dan gain di-update setiap 10 ms dan mereka dikalikan. Untuk kasus 2 kbps, hanya output dari stage pertama digunakan. Sedangkan untuk 4 kbps, output shape dari stage kedua dikalikan dengan output gain dari gain quantizer kedua dan ditambahkan ke waveform eksitasi dari stage pertama. Shape dan gain dari stage kedua di-update setiap 5 ms.



Gambar 7 – Decoder Time domain

### C4. Decoder Interpolasi Parameter

Decoder Interpolasi Parameter untuk kontrol kecepatan memiliki skema interpolasi parameter untuk menghasilkan parameter-parameter input time domain dan harmonic synthesizer pada saat waktu yang berubah-ubah. Dengan skema ini, deretan parameter dalam interval-interval termodifikasi dihitung dan dipergunakan untuk decoder source. Dalam hal ini, output decoder dalam skala waktu termodifikasi didapat. Karena parameter-parameter yang digunakan dengan mudah diinterpolasi, maka kompleksitas tambahan untuk modifikasi skala waktu sangat kecil.

Proses kontrol kecepatan dapat dijelaskan sebagai berikut. Rasio perubahan kecepatan didefinisikan sebagai  $spd$  :

$$spd = N_1 / N_2 \quad (6)$$

dimana  $N_1$  adalah durasi dari suara original dan  $N_2$  adalah durasi dari suara dengan kontrol kecepatan.

Untuk itu,

$$0 \leq n < N_1 \text{ dan } 0 \leq m < N_2$$

Parameter-parameter skala waktu termodifikasi dinyatakan sebagai berikut:

$$mdf\_param[m] = param[m \times spd] \quad (7)$$

dimana  $param$  adalah:  $pch$ ,  $vuv$ ,  $lsp$ , dan  $am$ . Namun, secara umum  $m \times spd$  bukan bilangan bulat.

Sehingga didefinisikan:

$$\begin{cases} fr_0 = \lceil m \times spd \rceil - 1 \\ fr_1 = fr_0 + 1 \end{cases} \quad (8)$$

untuk menghasilkan parameter-parameter pada indeks waktu  $m \times spd$  dengan interpolasi linier pada indeks-indeks waktu  $fr_0$  dan  $fr_1$ .

Untuk menjalankan interpolasi linier, definisikan:

$$\begin{cases} left = m \times spd - fr_0 \\ right = fr_1 - m \times spd \end{cases} \quad (9)$$

Kemudian persamaan (7) dapat dinyatakan sebagai:

$$mdf\_param[m] = param[fr_0] \times right + param[fr_1] \times left \quad (10)$$

dimana  $param$  adalah:  $pch$ ,  $vuv$ ,  $lsp$ , dan  $am$ .

### C5. Synthesizer Komponen Voiced

Synthesizer komponen voiced terdiri dari empat tahap yaitu sintesis eksitasi harmonik, penambahan komponen noise, sintesis LPC, dan postfilter. Metode sintesis eksitasi harmonik yang efisien pertama kali digunakan untuk mendapat waveform eksitasi periodik dari envelope magnituda harmonik. Komponen noise ditambahkan ke waveform periodik dan sinyal eksitasi voiced didapat, yang kemudian dimasukkan ke filter sintesis LPC dan postfilter untuk menghasilkan sinyal suara voiced. Konfigurasi postfilter tidak normative, bisa jadi dimodifikasi.

### C6. Synthesizer Komponen Unvoiced

Synthesizer komponen unvoiced tersusun dari filter sintesis LPC dan operasi postfilter. Untuk segmen unvoiced, skema VXC (CELP) digunakan. Koefisien-koefisien LPC yang hanya untuk *frame sekarang* digunakan untuk sub-frame tanpa interpolasi pada encoder dan decoder.

Sinyal eksitasi unvoiced yang dibangkitkan kemudian di-window untuk memudahkan (smoothly) dikoneksi ke sinyal voiced. Window untuk frame unvoiced ini digunakan hanya ketika frame unvoiced ditempatkan berdekatan dengan frame voiced atau yang sudah dicampur (mixed) dengan unvoiced.

Eksitasi unvoiced kemudian dimasukkan ke filter sintesis LPC. Output dari filter sintesis LPC kemudian diproses pada post filter yang terdiri dari tiga filter yaitu *high pass filter*, *high frequency emphasis filter*, dan *low pass filter*.

### C7. Alokasi Bit

Tabel 1 di bawah memperlihatkan alokasi bit dari frame HVXC 2.0 dan 4.0 kbps. Alokasi bit Common untuk mode *low delay* dan *normal delay*.

	V	Common	UV
LSF1		18bits/20msec	
LSF2		8bits/20msec	
V/UV		2bits/20msec	
pitch	7bits/20msec		
shape harmonik1	4+4bits/20msec		
gain harmonik1	5bits/20msec		
split harmonik2	32bits/20msec		
shape VXC1			6bits/10msec
gain VXC1			4bits/10msec
shape VXC2			5bits/5msec
gain VXC2			3bits/5msec
Total(1) 2kbps	40bits/20msec		40bits/20msec
Total(1&2) 4kbps	80bits/20msec		80bits/20msec

Tabel 1 – Alokasi bit HVXC

HVXC yang menggunakan bit rate 2 kbps memiliki frame (20ms) dengan alokasi bit yaitu 40 bit karena hanya output dari stage pertama yang digunakan. Frame Voiced (LSF1 + V/UV + pitch + shape harmonik1 + gain harmonik1 = 40 bit) dan frame Unvoiced (LSF1 + V/UV + shape VXC1 + gain VXC1 = 40 bit). HVXC dengan bit rate 4 kbps memiliki frame (20ms) dengan alokasi bit yaitu 80 bit karena output dari stage kedua juga digunakan. Frame Voiced (LSF1 + LSF2 + V/UV + pitch + shape harmonik1 + gain harmonik1 + split harmonik2 = 80 bit) dan frame Unvoiced (LSF1 + LSF2 + V/UV + shape VXC1 + gain VXC1 + shape VXC2 + gain VXC2 = 80 bit).

#### D. KESIMPULAN

Teknik pengkodean sinyal suara dengan Harmonic Vector eXcitation Coding (HVXC) berdasarkan pada pengkodean prediksi linier, namun menggunakan pendekatan yang berbeda untuk frame-frame voiced dan unvoiced. Komponen eksitasi unvoiced dinyatakan dalam vektor-vektor stochastic codebook. Komponen voiced dikodekan dalam bentuk spectral envelope sinyal eksitasi.

Hasil pengujian membuktikan pengkodean suara dengan HVXC memiliki efisiensi pengkodean sangat tinggi dengan faktor kompresi 64 pada 2 kbps yang lebih baik daripada CELP (FS-1016) pada 4.8 kbps [6].

#### E. UCAPAN TERIMA KASIH

Dalam penyelesaian tugas makalah ini penulis menerima bimbingan, saran, dorongan semangat dan bantuan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak DR. Ir. Suhartono T. sebagai Pembimbing Tugas Akhir MPEG-4 Speech Coding dengan HVXC.
2. Bapak Ir. Syafri Martinius sebagai Koordinator mata kuliah EL-400 Kerja Praktek dan Seminar.
3. Bapak Ir. Hamonangan, MT sebagai Koordinator mata kuliah EL-400 Kerja Praktek dan Seminar untuk Laboratorium Telematika.

#### F. DAFTAR PUSTAKA

- [1] ISO/IEC, “MPEG-4 Overview”, ISO/IEC JTC1/SC29/WG11 N3536, Beijing, July 2000.
- [2] ISO/IEC, “FDIS 14496-3: MPEG-4 Audio”, ISO/IEC JTC1/SC29/WG11 N2503, October 1998.
- [3] ISO/IEC, “FCD 0.1 14496-3 Subpart 2: Parametric Coding”, ISO/IEC JDC1/SC29/WG11 N2203, Maret 1998.
- [4] ISO/IEC, “MPEG-4 Applications”, ISO/IEC JTC1/SC-29/WG11 N2724, Seoul, Maret 1999.
- [5] ISO/IEC, “MPEG-4 Requirements, version 14 (Beijing revision)”, ISO/IEC JTC1/SC29/WG11 N3534, Beijing, July 2000.
- [6] Masayuki Nishiguchi, “MPEG-4 Speech Coding”, Audio and Speech Group, HomeNet Processing Lab, HomeNet Laboratories, Sony Corporation.
- [7] Jürgen Herre, Bernhard Grill, “Overview of MPEG-4 Audio and Its Applications in Mobile Communications”, Audio Department, Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany.
- [8] Heiko Purnhagen, “An Overview of MPEG-4 Audio Version 2”, Laboratorium für Informationstechnologie University of Hannover, Hannover, Germany.
- [9] Bernd Edler, “Very Low Bit Rate Audio Coding Development”, Laboratorium für Informationstechnologie University of Hannover, Germany.
- [10] <http://www.itiam.com/pages/products/hvxc.htm> : “Itiam Products Speech Codecs HVXC”.
- [11] <http://www.iis.fhg.de/amm/techinf/layer3/index.html> : “MPEG Audio Layer 3”.
- [12] <http://www.iis.fhg.de/amm/techinf/basics.html> : “Basics about MPEG Perceptual Audio Coding”.

#### G. BIOGRAFI



Made Santo Gitakarma, lahir di Denpasar pada tahun 1979. Menghabiskan pendidikan dasar dan menengah di Denpasar sampai tahun 1998. Kemudian melanjutkan studi ke jurusan Teknik Elektro Institut Teknologi Bandung dengan pilihan studi Telekomunikasi. Penelitian yang sedang dilakukan adalah MPEG-4 Speech Coding dengan HVXC.