

TEKNIK *QUADRATURE AMPLITUDE MODULATION (QAM)* UNTUK PENGIRIMAN DATA MELALUI FREKUENSI *VOICE BAND*

Abstrak

Teknik pengiriman data melalui frekuensi *voice band* merupakan teknik yang digunakan pada modem konvensional (modem *voice band*). Frekuensi *voice band* adalah frekuensi normal dari suara manusia yaitu dari 300 – 3000 Hz. Frekuensi ini merupakan frekuensi yang digunakan pada saluran telepon biasa.

Alasan dilakukan penelitian ini adalah untuk mengikuti perkembangan teknologi komputer dan jaringan yang berkembang pesat saat ini. Adapun tujuan penelitian adalah melakukan simulasi pengiriman data dengan frekuensi *voice band* menggunakan teknik *Quadrature Amplitude Modulation (QAM)* dengan prosesor sinyal digital.

Metodologi yang digunakan dalam simulasi pengiriman data ini meliputi perancangan pemancar dengan modulator QAM, perancangan penerima dengan carrier dan symbol clock synchronizer, demodulator QAM, serta pembuatan program untuk komputer pengirim data dan komputer penerima data. Selain itu ditambahkan pula root raised cosine pulse shaper pada pemancar dan penerima untuk mengurangi *intersymbol interference (ISI)* serta *adaptive equalizer* pada penerima untuk menghilangkan noise dan ISI yang terjadi. Saluran transmisi disimulasikan dengan menggunakan filter lolos rendah *Butterworth* dan *random noise generator* dengan tegangan rata-rata 0,3 V. Hasil yang didapatkan adalah sistem dapat mengirimkan simbol-simbol yang digunakan pada sistem dengan cukup baik. Sistem dapat dikembangkan lebih lanjut untuk pengiriman data yang lebih kompleks dengan memperbaiki beberapa bagian dari sistem khususnya *clock recovery* serta dengan menambahkan teknik *error correction*.

Pendahuluan

Modulasi adalah suatu proses penumpangan sinyal yang hendak dikirim pada sebuah sinyal *carrier*. Sinyal data dapat ditumpangkan ke sinyal *carrier* dengan cara mengubah amplitudo, frekuensi, atau fase dari sinyal *carrier* tersebut. Untuk mendapatkan laju pengiriman data yang lebih besar, dapat pula dilakukan perubahan terhadap kombinasi

dari beberapa parameter-parameter tersebut, misalnya dengan mengubah amplitudo dan fasenya sekaligus. Untuk pengiriman data melalui frekuensi *voiceband*, teknik yang biasanya digunakan adalah teknik modulasi digital multisimbol.

Teknik modulasi digital multisimbol ada beberapa macam, di antaranya adalah MPSK (*Multi-Phase Shift Keying*) dan QAM (*Quadrature Amplitude Modulation*). Modulasi digital multisimbol menawarkan kecepatan yang lebih tinggi karena setiap simbol yang dikirimkan melambangkan beberapa bit sekaligus. Untuk mendapat *data rate* yang tinggi pada *bandwidth* yang terbatas biasanya digunakan QAM.

Metodologi Penelitian

Tulisan ini membahas mengenai teknik modulasi QAM, perancangan *transmitter* dan *receiver* untuk mengirimkan data melalui frekuensi *voice band* menggunakan teknik modulasi QAM, menggunakan studi literatur terhadap buku-buku, majalah dan jurnal ilmiah serta eksperimen di laboratorium perangkat keras.

QUADRATURE AMPLITUDE MODULATION (QAM)

Pada QAM fase dan amplitudo dari sinyal *carrier* diubah-ubah untuk melambangkan data. Sinyal QAM dapat dituliskan sebagai berikut:

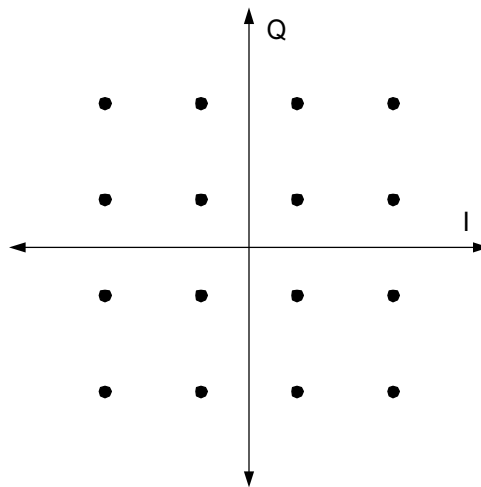
$$s(t) = I(t) \cdot \cos \omega_c t + Q(t) \cdot -\sin \omega_c t \quad (1)$$

dengan:

$$I(t) = A \cdot \cos \theta \quad (2)$$

$$Q(t) = A \cdot \sin \theta \quad (3)$$

Dari persamaan (1), dapat dilihat bahwa sinyal QAM dapat dibentuk dengan menjumlahkan sebuah sinyal kosinus dengan amplitudo $I(t)$ dan sebuah sinyal sinus dengan amplitudo $Q(t)$. Ini sama dengan menjumlahkan sebuah sinyal AM (*amplitude modulation*) yang menggunakan *carrier* kosinus dengan sebuah sinyal AM lain yang menggunakan *carrier* sinus. Kata *quadrature* pada QAM berasal dari kedua *carrier* yang berbeda fase 90° .



Gambar 1. Diagram Konstelasi 16-QAM

Amplitudo dan fase untuk masing-masing simbol pada QAM dapat digambarkan dalam sebuah diagram dua dimensi yang disebut sebagai diagram konstelasi, seperti misalnya diagram konstelasi untuk 16-QAM yang dapat dilihat pada gambar 1. Sumbu x merupakan sumbu yang mewakili $\cos \omega_c t$ dari persamaan (1) dan disebut sebagai sumbu I (*inphase*), sedangkan sumbu y adalah sumbu yang mewakili $-\sin \omega_c t$ dari persamaan (1) dan disebut sebagai sumbu Q (*quadrature*). Perlu diperhatikan

bahwa konfigurasi titik untuk 16-QAM tidak selalu seperti diagram konstelasi pada gambar 1.

Data yang akan dikirim dibagi menurut jumlah bit untuk satu simbol. Setelah itu, data yang telah dibagi dipetakan menurut diagram konstelasi dengan menggunakan *mapper*. Keluaran dari *mapper* adalah komponen *inphase* dan *quadrature* untuk simbol yang ditentukan oleh data tadi. Kedua komponen ini dijadikan sebagai masukan untuk modulator.

Pada penerima, sinyal yang telah dimodulasi perlu dikembalikan ke bentuk semula. Proses ini disebut sebagai proses demodulasi. Rumus untuk demodulator adalah sebagai berikut:

$$I'(t) = s(t) \cdot \cos \omega_c' t - s_h(t) \cdot \sin \omega_c' t \quad (4)$$

$$Q'(t) = s(t) \cdot \sin \omega_c' t + s_h(t) \cdot \cos \omega_c' t \quad (5)$$

$s(t)$ adalah sinyal yang diterima sedangkan $s_h(t)$ dihasilkan dengan melakukan *Hilbert transform* terhadap $s(t)$. ω_c' adalah frekuensi *carrier* yang dibangkitkan pada penerima.

Hasil dari demodulator dimasukkan ke *decision unit* yang merupakan unit untuk menentukan simbol yang telah dikirim oleh pemancar dari hasil demodulasi. *Decision unit* menentukan simbol mana yang sebenarnya dikirim oleh pemancar dengan cara menghitung jarak Euclidean dari titik yang diterima dengan semua titik yang ada pada diagram konstelasi (Massey, 1997). Simbol yang dipilih adalah simbol yang jaraknya paling kecil dengan titik yang diterima. Keluaran dari *decision unit* adalah komponen *inphase* dan *quadrature* dari titik ideal untuk simbol yang dipilih. Keluaran ini dikembalikan ke dalam bentuk bit-bit oleh unit *inverse mapper*.

Pada proses demodulasi diperlukan *carrier* lokal yang memiliki frekuensi dan fase yang sama dengan *carrier* dari sinyal yang diterima. Perbedaan frekuensi *carrier* pemancar dengan frekuensi *carrier* penerima dapat terjadi karena adanya perbedaan frekuensi dari *oscillator* yang digunakan pada pemancar dengan yang digunakan pada penerima. Perbedaan frekuensi dari *oscillator* disebabkan karena ketidakakuratan pada pembuatannya, perbedaan temperatur, dan lain-lain. Perbedaan fase terjadi terutama pada saat pemancar dan penerima pertama kali dihubungkan dan *delay* fase pada sinyal yang terjadi pada saluran transmisi.

Untuk mendapatkan sinyal *carrier* yang sinkron dengan sinyal *carrier* pada pemancar, maka diperlukan sinkronisasi *carrier*. Sinkronisasi *carrier* dapat dilakukan dengan mengekstrak sinyal *carrier* dari sinyal data yang diterima dengan menggunakan metode sinkronisasi *carrier* seperti *squaring loop*, *costas loop*, atau *decision feedback loop*. Di antara ketiga macam sinkronisasi *carrier* tersebut yang terbaik adalah *decision feedback loop*. Metode ini menggunakan hasil dari *decision unit* yang bebas dari *noise* untuk menghitung kesalahan fase dari *carrier* yang dibangkitkan (Proakis, 1995).

Pada metode *decision feedback loop* digunakan unit *phase-locked loop* (PLL). Bagian dari PLL adalah *phase detector*, *loop filter*, dan *voltage controlled oscillator* (VCO). *Phase detector* berfungsi untuk menghitung perbedaan fase dari kedua *carrier*. *Loop filter* berfungsi untuk meredam *noise* yang masuk dan VCO berfungsi untuk membangkitkan sinyal *carrier* lokal.

Pada *phase detector*, *error* fase dihitung dengan menggunakan rumus:

$$E(t) = QDEC(t) \cdot IDEMOD(t) - IDEC(t) \cdot QDEMOD(t) \quad (6)$$

Perbedaan frekuensi antara *carrier* lokal dengan *carrier* pada penerima dapat dikompensasi dengan perbaikan terhadap fase *carrier* karena besarnya relatif sangat kecil.

Pada awal pengiriman data, kedua *carrier* belum sinkron sehingga hasil demodulasi akan salah. Karena hasil demodulasi salah, maka hasil *decision* juga akan salah. Hasil *decision* yang salah akan menyebabkan kesalahan juga pada *carrier recovery* sehingga *carrier* tidak dapat disinkronisasi. Karena *carrier* tidak dapat disinkronisasi, maka hasil demodulasi yang selanjutnya akan salah lagi. Hal ini akan terjadi terus-menerus sehingga pengiriman data akan mengalami kegagalan.

Untuk mengatasi masalah ini, maka pada saat pertama kali pemancar dan penerima berhubungan, dilakukan suatu fase awal yang disebut sebagai fase *training*. Di dalam fase ini, pemancar mengirimkan suatu sinyal yang sudah diketahui oleh penerima. Pada fase ini, *decision unit* pada penerima tidak diaktifkan dan komponen *inphase* dan *quadrature* yang seharusnya berasal dari unit tersebut dipaksa dengan nilai yang benar sesuai dengan sinyal yang dikirimkan oleh pemancar. Dengan nilai tersebut, maka hasil *decision* menjadi benar selama fase *training*.

Dengan hasil *decision* yang benar, maka *carrier recovery* dapat bekerja dan secara bertahap menghasilkan *carrier* yang benar. Lamanya fase *training* harus cukup bagi *carrier recovery* untuk menghasilkan *carrier* lokal yang sefase dengan *carrier* dari sinyal yang diterima. Setelah fase *training* selesai, maka kedua *carrier* sudah sefase dan *decision unit* dapat diaktifkan.

Penerima juga memerlukan unit *clock recovery* untuk mendeteksi pertengahan dari simbol yang dikirim (Troullinos, 1997). Jika sudah terdeteksi, maka *decision unit* dapat membuat keputusan dengan probabilitas kesalahan yang paling kecil. Hal ini disebabkan karena energi dari sinyal yang diterima besarnya maksimum pada pertengahan simbol. Selain itu, filter *raised cosine* mengatur agar *zero interference intersymbol* (ISI) terjadi pada pertengahan simbol.

Demodulasi QAM

Pada penerima, sinyal yang telah dimodulasi perlu dikembalikan ke bentuk semula. Proses ini disebut sebagai proses demodulasi. Demodulasi yang umum digunakan untuk QAM adalah demodulasi yang bersifat sinkronus. Berikut ini akan diuraikan penurunan rumus demodulasi untuk QAM (Troullinos, 1997). Untuk menurunkan rumus demodulasi, maka perlu dilakukan tinjauan terhadap proses modulasi-demodulasi dengan menggunakan bilangan kompleks. Persamaan (2.5) dan (2.6) dapat dituliskan dalam bentuk:

$$c(t) = I(t) + jQ(t) \quad (7)$$

$c(t)$ dimodulasi dengan menggeser spektrumnya sejauh f_c (frekuensi carrier). Untuk menggeser spektrum dari suatu sinyal, maka bentuk *time domain* dari sinyal tersebut harus dikalikan dengan $e^{j\phi t}$, di mana ϕ adalah besar pergeseran yang diinginkan. Dalam kasus ini, ϕ adalah ω_c atau $2\pi f_c$ sehingga persamaan untuk sinyal yang telah dimodulasi adalah sebagai berikut:

$$m(t) = c(t).e^{j\omega_c t} \quad (8)$$

Persamaan (2.7) disubstitusi ke persamaan (2.8) menjadi:

$$m(t) = \{I(t) + Q(t)\}.e^{j\omega_c t} \quad (9)$$

Kemudian dengan menggunakan rumus identitas euler:

$$e^{j\omega_c t} = \cos \omega_c t + j.\sin \omega_c t \quad (10)$$

maka didapat:

$$m(t) = \{I(t).\cos \omega_c t + Q(t).-\sin \omega_c t\} + j.\{-I(t).-\sin \omega_c t + Q(t).\cos \omega_c t\} \quad (11)$$

Bagian real dan imajiner dari persamaan (2.11) merupakan suatu pasangan *Hilbert transform*. Dua sinyal disebut sebagai pasangan *Hilbert transform* jika keduanya dihubungkan dengan sebuah *Hilbert transform*. *Hilbert transform* diimplementasikan dengan sebuah filter yang disebut sebagai *Hilbert transformer*. Sifat *Hilbert transform* yang menghubungkan bagian real dan imajiner dari persamaan (11) memungkinkan pengiriman bagian real saja dari persamaan (11) untuk menghemat *bandwidth*. Bagian imajinernya dibentuk kembali pada penerima dengan melakukan *Hilbert transform* terhadap sinyal yang diterima.

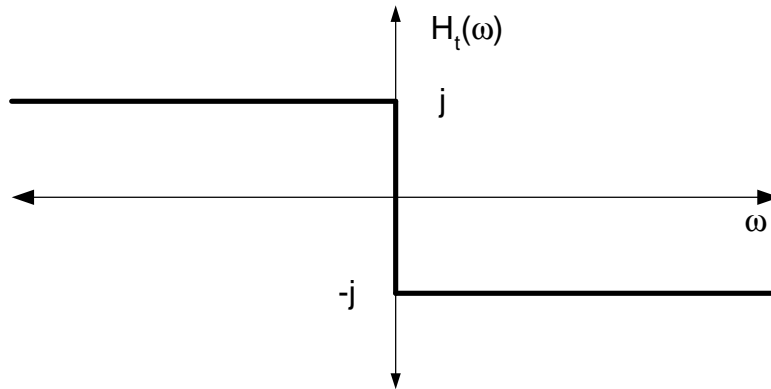
Sebuah *Hilbert transformer* didefinisikan sebagai filter dengan fungsi transfer sebagai berikut:

$$H_t(\omega) = -e^{j\frac{\pi}{2}} \text{sgn}(\omega) = -j.\text{sgn}(\omega) \quad (12)$$

di mana *sgn* adalah fungsi *sign* yang didefinisikan sebagai berikut:

$$\text{sgn}(\omega) = \begin{cases} +1, & \text{jika } \omega > 0 \\ -1, & \text{jika } \omega < 0 \end{cases} \quad (13)$$

Karakteristik fungsi transfer dari *Hilbert transformer* ditunjukkan pada gambar 2.2. pada gambar tersebut dapat dilihat bahwa *Hilbert transformer* akan menimbulkan pergeseran fase sebesar -90° untuk frekuensi positif ($\omega > 0$) dan pergeseran fase sebesar $+90^\circ$ untuk frekuensi negatif ($\omega < 0$).



Gambar 2. Fungsi transfer *Hilbert transformer*

Hubungan pasangan *Hilbert transform* antara bagian real dan imajiner dari persamaan (11) akan ditunjukkan sebagai berikut: Anggap bahwa $S(\omega)$ dan $S_h(\omega)$ masing-masing adalah transformasi fourier dari bagian real dan imajiner pada persamaan (11) sehingga:

$$S(\omega) = \frac{1}{2}\{I(\omega + \omega_c) + I(\omega - \omega_c)\} - \frac{j}{2}\{Q(\omega + \omega_c) - Q(\omega - \omega_c)\} \quad (14)$$

$$S_h(\omega) = \frac{j}{2}\{I(\omega + \omega_c) - I(\omega - \omega_c)\} + \frac{1}{2}\{Q(\omega + \omega_c) + Q(\omega - \omega_c)\} \quad (15)$$

di mana $I(\omega)$ dan $Q(\omega)$ masing-masing adalah transformasi fourier dari $I(t)$ dan $Q(t)$.

Dengan $S(\omega)$ sebagai masukan bagi *Hilbert transformer*, maka keluarannya dalam domain frekuensi adalah:

$$O(\omega) = S(\omega) \cdot H(\omega) = -j \cdot S(\omega) \cdot \text{sgn}(\omega) \quad (16)$$

Substitusi persamaan (2.14) ke dalam persamaan (16) menghasilkan:

$$O(\omega) = -j \cdot \left\{ \frac{1}{2} \{I(\omega + \omega_c) + I(\omega - \omega_c)\} - \frac{j}{2} \{Q(\omega + \omega_c) - Q(\omega - \omega_c)\} \right\} \cdot \text{sgn}(\omega)$$

$$O(\omega) = \left\{ -\frac{j}{2} \{I(\omega + \omega_c) + I(\omega - \omega_c)\} - \frac{1}{2} \{Q(\omega + \omega_c) - Q(\omega - \omega_c)\} \right\} \cdot \text{sgn}(\omega)$$

$$O(\omega) = \frac{j}{2} \{I(\omega + \omega_c) - I(\omega - \omega_c)\} + \frac{1}{2} \{Q(\omega + \omega_c) + Q(\omega - \omega_c)\} \quad (17)$$

Persamaan (17) sama dengan persamaan (15) sehingga terbukti bahwa bagian imajiner dari persamaan (11) merupakan pasangan *Hilbert transform* dari bagian realnya.

Persamaan (11) dapat dituliskan sebagai berikut:

$$m(t) = s(t) + js_h(t) \quad (18)$$

Pada penerima, mendapatkan kembali bagian imajiner $s_h(t)$ dilakukan dengan melakukan *Hilbert transform* terhadap bagian real $s(t)$ (sinyal yang diterima).

Persamaan untuk demodulator diturunkan sebagai berikut: Demodulator mengembalikan sinyal yang telah dimodulasi ke bentuk semula dengan mengalikan sinyal tersebut dengan $e^{-j\omega_c t}$:

$$c'(t) = m(t) \cdot e^{-j\omega_c t} \quad (19)$$

dengan ω_c' sama dengan frekuensi *carrier* yang dibangkitkan pada penerima. Anggap $\omega_c' = \omega_c$ sehingga dengan mensubstitusikan persamaan (8) ke persamaan (19), maka didapatkan:

$$c'(t) = c(t) \cdot e^{j\omega_c t} \cdot e^{-j\omega_c t} = c(t) \quad (20)$$

Jika $I'(t)$ dan $Q'(t)$ adalah bagian real dan imajiner dari $c'(t)$ dan dengan melihat persamaan (2.7), maka:

$$I'(t) = I(t) \quad (21)$$

$$Q'(t) = Q(t) \quad (22)$$

Rumus yang digunakan di demodulator didapatkan dengan mensubstitusi persamaan (2.18) ke persamaan (2.19) dan menggunakan rumus identitas euler:

$$c'(t) = \{s(t).\cos \omega_c t - s_h(t).-\sin \omega_c t\} + j.\{s(t).-\sin \omega_c t + s_h(t).\cos \omega_c t\} \quad (23)$$

Dengan menggunakan bagian real dan imajiner dari persamaan (21), maka didapatkan rumus untuk demodulator sebagai berikut:

$$I'(t) = s(t).\cos \omega_c t - s_h(t).-\sin \omega_c t \quad (24)$$

$$Q'(t) = s(t).-\sin \omega_c t + s_h(t).\cos \omega_c t \quad (25)$$

Hasil dari demodulator dimasukkan ke *decision unit* yang merupakan unit untuk menentukan simbol yang telah dikirim oleh pemancar dari hasil demodulasi. Hasil demodulasi memang secara teori sudah merupakan komponen *inphase* dan *quadrature* yang dikirimkan oleh pemancar. Namun faktor *noise*, *intersymbol interference*, dan gangguan-gangguan lainnya, akan membuat komponen *inphase* dan *quadrature* tersebut tidak sama besarnya dengan komponen *inphase* dan *quadrature* yang ideal. Ini akan menyebabkan posisi titik dari simbol yang diterima tidak sama dengan posisi titik yang ideal berdasarkan diagram konstelasi.

Decision Unit

Decision unit menentukan simbol mana yang sebenarnya dikirim oleh pemancar dengan cara menghitung jarak Euclidean dari titik yang diterima dengan semua titik yang ada pada diagram konstelasi. Jarak Euclidean adalah jarak dalam konteks ruang

yang dihitung dengan rumus sebagai berikut: Jika titik yang diterima memiliki koordinat $(I'(t), Q'(t))$ dan suatu titik ideal memiliki koordinat (x, y) , maka jarak Euclidean-nya adalah:

$$E = \sqrt{\{y - Q'(t)\}^2 + \{x - I'(t)\}^2} \quad (26)$$

Simbol yang dipilih adalah simbol yang jaraknya paling kecil dengan titik yang diterima. Keluaran dari *decision unit* adalah komponen *inphase* dan *quadrature* dari titik ideal untuk simbol yang dipilih. Keluaran ini dikembalikan ke dalam bentuk bit-bit oleh unit *inverse mapper*.

Carrier Recovery

Dalam proses demodulasi, sinyal yang datang terlebih dahulu dilewatkan ke *Hilbert transformer* kemudian dikalikan dengan sinyal *carrier* sinus atau kosinus yang dibangkitkan pada penerima sesuai dengan persamaan (23) dan (24). Dalam penurunan rumus untuk demodulasi di atas, digunakan asumsi bahwa frekuensi *carrier* pada penerima sama dengan frekuensi *carrier* pada pemancar. Dari persamaan (20) dapat dilihat jika frekuensi dan fase dari sinyal *carrier* pada penerima tidak sama dengan frekuensi dan fase dari sinyal *carrier* pada pemancar, maka hasil demodulasi tidak sama dengan $c(t)$.

Perbedaan frekuensi *carrier* pemancar dengan frekuensi *carrier* penerima dapat terjadi karena adanya perbedaan frekuensi dari *oscillator* yang digunakan pada pemancar dengan yang digunakan pada penerima. Perbedaan frekuensi dari *oscillator* disebabkan karena ketidakakuratan pada pembuatannya, perbedaan temperatur, dan lain-lain. Perbedaan fase terjadi terutama pada saat pemancar dan penerima pertama kali dihubungkan. Anggap pemancar dan penerima diatur agar pada saat dinyalakan sinyal

carrier dimulai dari 0° . Dengan menyalakan kedua sistem secara bersamaan, maka kedua sinyal *carrier* akan sefase. Namun tentu saja hal ini tidak mungkin dilakukan dan meskipun ada toleransi untuk perbedaan fase, metode ini sangat tidak praktis.

Jika gangguan pada sinyal yang diterima juga dipertimbangkan, maka akan terjadi *delay* fase pada sinyal sehingga frekuensi dan fase sinyal yang diterima tidak sama dengan frekuensi dan fase dari sinyal yang dikirim oleh pemancar. Oleh sebab itu, frekuensi dan fase dari sinyal *carrier* yang dibangkitkan pada penerima harus sama dengan frekuensi dan fase dari sinyal yang diterima, bukan dengan sinyal pada pemancar.

Seperti yang telah disebutkan sebelumnya, demodulasi yang digunakan adalah demodulasi sinkronus. Dengan penurunan rumus untuk demodulasi di atas, maka maksud dari istilah sinkronus pada demodulasi menjadi jelas, yaitu proses demodulasi memerlukan *carrier* yang sinkron dengan *carrier* yang diterima (frekuensi dan fasenya sama).

Untuk mendapatkan sinyal *carrier* yang sinkron dengan sinyal *carrier* pada pemancar, maka diperlukan sinkronisasi *carrier*. Sinkronisasi *carrier* dapat dilakukan dengan mengirimkan *carrier* bersama dengan sinyal yang telah dimodulasi menggunakan *frequency division multiplexing* (FDM). *Carrier* tersebut diekstrak pada penerima dengan menggunakan *phase-locked loop* (PLL). Cara ini lebih mudah namun membutuhkan energi dan *bandwidth* yang lebih besar.

Cara yang lebih efisien adalah dengan mengekstrak sinyal *carrier* dari sinyal data yang diterima. *Carrier* diekstrak dari sinyal data juga dengan menggunakan PLL, namun secara keseluruhan unit sinkronisasi *carrier* tidak hanya terdiri dari PLL karena *carrier* harus dipisahkan dahulu dari sinyal data. Terdapat beberapa macam metode

sinkronisasi *carrier*, antara lain *squaring loop*, *costas loop*, dan *decision feedback loop*. Di antara ketiga macam sinkronisasi *carrier* tersebut yang terbaik adalah *decision feedback loop*. Metode ini menggunakan hasil dari *decision unit* yang bebas dari *noise* untuk menghitung kesalahan fase dari *carrier* yang dibangkitkan. Hasil *decision* dapat dikatakan bebas dari *noise* karena keluarannya adalah berupa komponen *inphase* dan *quadrature* dari titik ideal. (Proakis, 1995)

Pengaruh dari perbedaan fase (Troullinos, 1997) dapat dilihat sebagai berikut:

Persamaan (2.11) dapat dituliskan sebagai berikut:

$$s(t) = I(t) \cdot \cos(\omega_c t + \theta_r) + Q(t) \cdot \sin(\omega_c t + \theta_r) \quad (27)$$

$$s_h(t) = -I(t) \cdot \sin(\omega_c t + \theta_r) + Q(t) \cdot \cos(\omega_c t + \theta_r) \quad (28)$$

dengan ω_c dan θ_r sebagai frekuensi dan fase dari sinyal *carrier* yang diterima. Hasil demodulasi didapat dengan persamaan (24) dan (25) dan dengan menambahkan θ_l sebagai fase dari sinyal *carrier* lokal, maka hasil demodulasi dapat dituliskan kembali sebagai:

$$IDEMOD(t) = s(t) \cdot \cos(\omega_c' t + \theta_l) - s_h(t) \cdot \sin(\omega_c' t + \theta_l) \quad (29)$$

$$QDEMOD(t) = s(t) \cdot \sin(\omega_c' t + \theta_l) + s_h(t) \cdot \cos(\omega_c' t + \theta_l) \quad (30)$$

di mana ω_c' adalah frekuensi *carrier* lokal.

Dengan menganggap $\omega_c = \omega_c'$, maka substitusi persamaan (27) dan (28) dengan persamaan (2.29) dan (2.30) akan mendapatkan:

$$IDEMOD(t) = I(t) \cdot \cos \theta_e - Q(t) \cdot \sin \theta_e \quad (31)$$

$$QDEMOD(t) = I(t) \cdot \sin \theta_e + Q(t) \cdot \cos \theta_e \quad (32)$$

dimana $\theta_e = \theta_r - \theta_l$ adalah beda fase antara kedua *carrier*. Jika $\theta_e = 0$, maka:

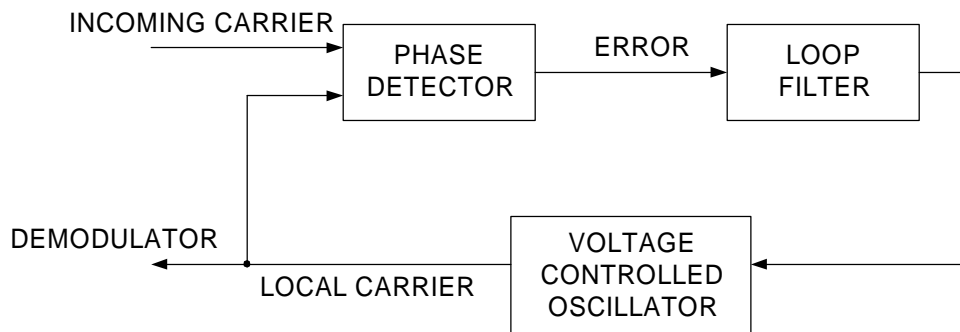
$$IDEMOD(t) = I(t) \quad (33)$$

$$Q_{\text{DEMOD}}(t) = Q(t) \quad (34)$$

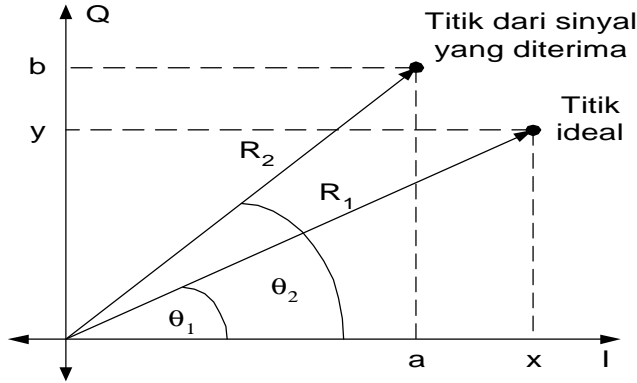
Persamaan (31) dan (32) menunjukkan jika terdapat perbedaan fase antara kedua *carrier*, maka akan terjadi pelemahan (karena $\cos \theta_e \neq 0$) dan *crosstalk* antara komponen *inphase* dengan komponen *quadrature* (karena $\sin \theta_e \neq 0$). Sedangkan persamaan (33) dan (34) menunjukkan jika tidak ada perbedaan fase antara kedua *carrier*, maka hasil demodulasi sama dengan komponen *inphase* dan *quadrature* yang dikirimkan.

Phase-Locked Loop pada Decision Feedback Loop

Berikut ini akan dijelaskan mengenai PLL yang digunakan untuk *decision feedback loop*. Blok diagram dari PLL ini ada pada gambar 3. Bagian dari PLL adalah *phase detector*, *loop filter*, dan *voltage controlled oscillator (VCO)*. *Phase detector* berfungsi untuk menghitung perbedaan fase dari kedua *carrier*. *Loop filter* berfungsi untuk meredam *noise* yang masuk dan VCO berfungsi untuk membangkitkan sinyal *carrier* lokal.



Gambar 3. Phase-locked loop untuk carrier recovery



Gambar 4. Titik yang diterima dan titik idealnya

Pada *phase detector*, untuk menghitung *error* fase antara *carrier* lokal dengan *carrier* yang diterima, digunakan rumus yang akan diturunkan sebagai berikut: Anggap bahwa frekuensi *carrier* lokal sama dengan frekuensi *carrier* dari sinyal yang diterima. Dengan menggunakan hasil *decision*, maka perbedaan fase kedua *carrier* dapat dilihat dari perbedaan fase antara titik pada diagram konstelasi untuk sinyal yang diterima dengan titik ideal yang seharusnya diterima jika tidak ada kesalahan atau gangguan apapun. Ini dapat dilihat pada gambar 4. Dari gambar 4 dapat dituliskan:

$$\sin(\theta_1 - \theta_2) = \sin \theta_1 \cdot \cos \theta_2 - \sin \theta_2 \cdot \cos \theta_1$$

$$\sin(\theta_1 - \theta_2) = \frac{y}{R_1} \cdot \frac{a}{R_2} - \frac{b}{R_2} \cdot \frac{x}{R_1} = \frac{ya - xb}{R_1 R_2} \quad (35)$$

Sinyal error didapatkan dari:

$$E(t) = R_1 R_2 \cdot \sin(\theta_1 - \theta_2) = ya - xb \quad (36)$$

atau dengan menggunakan $a = \text{IDEMOD}(t)$ dan $b = \text{QDEMOD}(t)$ sebagai hasil demodulasi, $x = \text{IDEC}(t)$ dan $y = \text{QDEC}(t)$ sebagai hasil *decision*, maka didapatkan rumus untuk *phase detector*:

$$E(t) = \text{QDEC}(t) \cdot \text{IDEMOD}(t) - \text{IDEC}(t) \cdot \text{QDEMOD}(t) \quad (37)$$

Jika perbedaan frekuensi juga dihitung, maka akan didapatkan (Troullinos, 1997):

$$E(t) = R_1 R_2 \cdot \sin(\omega_e t + \theta_e) \quad (38)$$

di mana $\omega_e = \omega - \omega'$ adalah perbedaan frekuensi antara kedua *carrier*. Namun perbedaan frekuensi ini sangat kecil sehingga dapat dikompensasi dengan perbaikan terhadap fase *carrier*. Jadi dengan menggunakan persamaan (37), perbedaan fase dan frekuensi dapat diminimalkan.

Dari persamaan (36) dapat dilihat jika *error* fase positif, maka $E(t)$ akan positif sedangkan jika *error* fase negatif, maka $E(t)$ akan negatif. Jika *error* fase positif, maka *carrier* lokal harus dimajukan (*advanced*) sedangkan jika *error* fase negatif, maka *carrier* lokal harus ditunda (*retarded*).

Fase Training

Pada awal pengiriman data, kedua *carrier* belum sinkron sehingga hasil demodulasi akan salah. Karena hasil demodulasi salah, maka hasil *decision* juga akan salah. Hasil *decision* yang salah akan menyebabkan kesalahan juga pada *carrier recovery* sehingga *carrier* tidak dapat disinkronisasi. Karena *carrier* tidak dapat disinkronisasi, maka hasil demodulasi yang selanjutnya akan salah lagi. Hal ini akan terjadi terus-menerus sehingga pengiriman data akan mengalami kegagalan.

Untuk mengatasi masalah ini, maka pada saat pertama kali pemancar dan penerima berhubungan, dilakukan suatu fase awal yang disebut sebagai fase *training*. Di dalam fase ini, pemancar mengirimkan suatu sinyal yang sudah diketahui oleh penerima. Pada fase ini, *decision unit* pada penerima tidak diaktifkan dan komponen *inphase* dan *quadrature* yang seharusnya berasal dari unit tersebut dipaksa dengan nilai yang benar sesuai dengan sinyal yang dikirimkan oleh pemancar. Dengan nilai tersebut, maka hasil *decision* menjadi benar selama fase *training*.

Dengan hasil *decision* yang benar, maka *carrier recovery* dapat bekerja dan secara bertahap menghasilkan *carrier* yang benar. Lamanya fase *training* harus cukup bagi *carrier recovery* untuk menghasilkan *carrier* lokal yang sefase dengan *carrier* dari sinyal yang diterima. Setelah fase *training* selesai, maka kedua *carrier* sudah sefase dan *decision unit* dapat diaktifkan.

Karena adanya *delay* fase yang ditimbulkan oleh saluran transmisi, maka fase dari sinyal yang diterima akan terdistorsi. Akibatnya fase dari sinyal *carrier* yang diterima dapat berubah-ubah. Oleh sebab itu, walaupun fase dari *carrier* lokal dengan *carrier* dari sinyal yang diterima sudah disamakan pada saat *training*, *carrier recovery* harus terus dijalankan untuk menyamakan fase setelah *training* selesai.

Clock Recovery

Penerima juga memerlukan unit *clock recovery* untuk mendeteksi pertengahan dari simbol yang dikirim (Troullinos, 1997). Jika sudah terdeteksi, maka *decision unit* dapat membuat keputusan dengan probabilitas kesalahan yang paling kecil. Hal ini disebabkan karena energi dari sinyal yang diterima besarnya maksimum pada pertengahan simbol. Selain itu, filter *raised cosine* mengatur agar *zero interference intersymbol* (ISI) terjadi pada pertengahan simbol.

Proses *clock recovery* dilakukan dengan menghitung jumlah dari kuadrat komponen *inphase* dan *quadrature*. Hasil dari penjumlahan tersebut dilewatkan pada sebuah BPF dengan *bandwidth* yang sangat sempit (*resonator*). Frekuensi *center* dari BPF tersebut dibuat sebesar frekuensi simbol yang digunakan. Keluaran dari BPF adalah sebuah sinyal sinus yang memiliki frekuensi sama dengan frekuensi simbol. Sinyal sinus

tersebut dilewatkan pada sebuah unit yang mendeteksi pertengahan dari sinyal sinus tersebut.

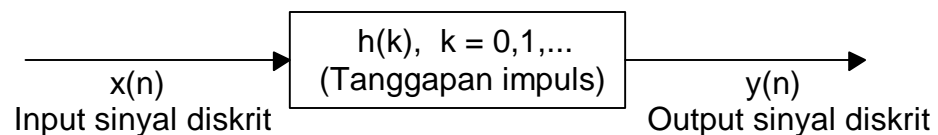
Filter Digital

Filter adalah sebuah sistem atau jaringan yang secara selektif merubah karakteristik (bentuk gelombang, frekuensi, fase dan amplitudo) dari sebuah sinyal. Secara umum tujuan dari pemfilteran adalah untuk meningkatkan kualitas dari sebuah sinyal sebagai contoh untuk menghilangkan atau mengurangi *noise*, mendapatkan informasi yang dibawa oleh sinyal atau untuk memisahkan dua atau lebih sinyal yang sebelumnya dikombinasikan, dimana sinyal tersebut dikombinasikan dengan tujuan mengefisienkan pemakaian saluran komunikasi yang ada.

Filter digital adalah sebuah implementasi algoritma matematik ke dalam perangkat keras dan/atau perangkat lunak yang beroperasi pada sebuah input sinyal digital untuk menghasilkan sebuah output sinyal digital agar tujuan pemfilteran tercapai. Filter digital memainkan peranan yang sangat penting dalam pemrosesan sinyal digital, pemakaian filter digital antara lain untuk kompresi data, pemrosesan suara, pengolahan citra, pengiriman data dan *echo canceller* pada telepon.

Filter Digital Finite Impuls Response dan Infinite Impuls Response

Diagram blok dasar dari filter digital dapat dilihat pada gambar 5, berdasarkan diagram blok tersebut maka secara garis besar filter digital dapat dibagi menjadi dua yaitu filter digital dengan tanggapan impuls berhingga (*FIR*) dan filter digital dengan tanggapan impuls tak berhingga (*IIR*).



Gambar 5 Blok Diagram Dasar Filter Digital

Filter digital *FIR* dan *IIR* dapat dituliskan dalam bentuk persamaan :

$$FIR : \quad y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (39)$$

$$IIR : \quad y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (40)$$

Dari persamaan (39) terlihat bahwa filter *FIR* memiliki durasi yang berhingga sedangkan filter *IIR* memiliki durasi yang tak berhingga. Secara praktis filter *FIR* dapat langsung dibuat dengan menggunakan persamaan (39), sedangkan untuk filter *IIR* tidak mungkin dibuat dengan menggunakan persamaan (40) sebab memerlukan memori yang tak berhingga sehingga agar secara praktis dapat dibuat maka persamaan untuk *IIR* dirubah menjadi bentuk rekursif yaitu :

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^N a_k x(n-k) - \sum_{k=0}^M b_k y(n-k) \quad (41)$$

dimana a_k dan b_k adalah koefisien filter. Persamaan (41) dikatakan rekursif sebab output dari filter *IIR* pada suatu saat bergantung dengan output yang sebelumnya.

Persamaan fungsi alih untuk filter *FIR* dan *IIR* dinyatakan dengan menggunakan Z-Transform yaitu :

$$FIR : \quad H(z) = \sum_{k=0}^{N-1} h(k)z^{-k} \quad (42)$$

$$IIR : \quad H(z) = \frac{\sum_{k=0}^N a_k z^{-k}}{(1 + \sum_{k=1}^M b_k z^{-k})} \quad (43)$$

Filter Digital Adaptif

Filter digital adaptif merupakan filter digital yang mampu melakukan pengaturan terhadap koefisien-koefisiennya secara otomatis. Algoritma adaptif yang banyak digunakan adalah algoritma *Least Mean Square (LMS)* karena komputasi dan tempat penyimpanan yang dibutuhkan lebih efisien. Algoritma *LMS* bertujuan untuk meminimalkan kuadrat dari error yang terjadi. Persamaan (44) adalah rumus untuk memperbaharui koefisien setiap pencuplikan pada algoritma *LMS*.

$$W_{n+1} = W_n + 2m e_n X_n \quad (44)$$

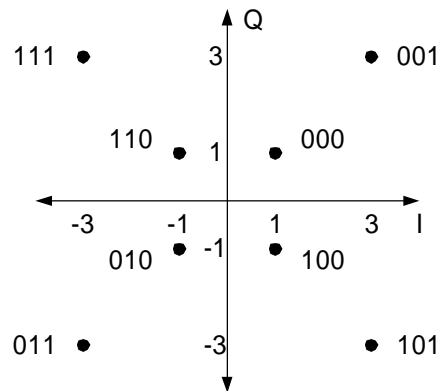
Rumus untuk memperbaharui koefisien pada algoritma *LMS* persamaan (44) relatif sederhana sebab hanya melibatkan operasi perkalian dan operasi penjumlahan sehingga mudah dilakukan pada operasi *real time*. Untuk nilai awal tertentu dari koefisien algoritma *LMS* akan konvergen dan stabil jika :

$$0 < m < \frac{1}{I_{\max}} \quad (45)$$

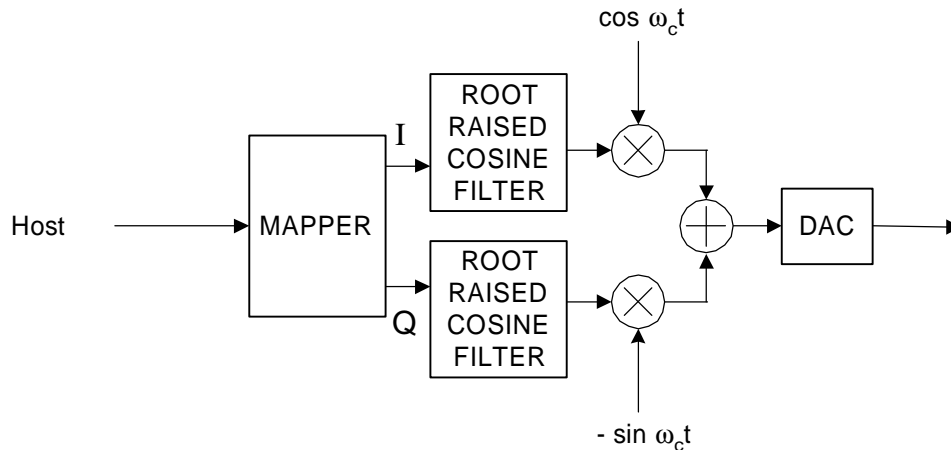
dimana $I_{\max} = N \cdot \text{daya}$ dari sinyal input atau maksimum nilai *eigen* dari data input kovarian matriks. Secara praktis, W_n tak pernah mencapai nilai optimum secara teori tapi berfluktuasi di sekitar titik itu.

ANALISIS

Pemrosesan sinyal yang dilakukan untuk pengiriman data ini diatur seluruhnya dengan menggunakan program untuk prosesor DSP. Program untuk DSP dibuat dengan menggunakan *algebraic assembler* yang disediakan untuk DSKplus TMS320C542.



Gambar 6 Diagram Konstelasi Sistem



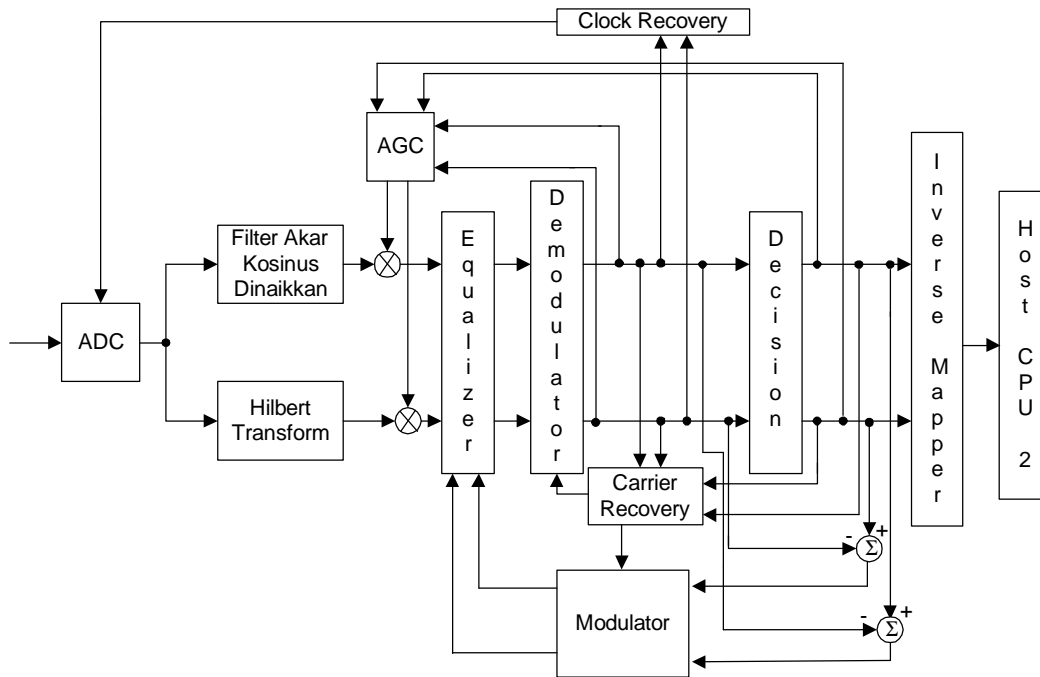
Gambar 7 Diagram Blok Pemancar

Sistem menggunakan modulasi 8-QAM dengan frekuensi *carrier* 1736,11 Hz. Frekuensi *sampling* yang digunakan adalah 9259,259 Hz dan kecepatan simbol sebesar 2314,815 baud. Karena modulasi yang digunakan adalah 8-QAM, maka satu simbol mewakili 3 bit sehingga besarnya kecepatan transmisi data adalah 6944,445 bps. Diagram konstelasi yang digunakan pada sistem ada pada gambar 6 sedangkan diagram blok pemancar ada pada gambar 7.

Simbol dikirim dari *host* melalui port paralel ke HPI (*Host Port Interface*) dari DSP. Banyaknya simbol yang dikirim oleh *host* dalam satu pengiriman adalah 3 simbol yang dirangkai menjadi 9 bit. DSP membagi rangkaian 9 bit tersebut menjadi simbol-simbol yang masing-masing terdiri dari 3 bit. Kemudian 3 bit ini dipetakan oleh *mapper* untuk menentukan komponen *inphase* dan *quadrature*-nya. *Mapper* diimplementasikan dengan menggunakan *look-up table*.

Komponen *inphase* dan *quadrature* keluaran dari *mapper* difilter terlebih dahulu dengan filter *baseband root raised cosine* untuk mengurangi efek *intersymbol interference* (ISI) yang muncul akibat terbatasnya *bandwidth* saluran. Kemudian keluaran dari filter dimodulasi dengan *carrier* kosinus dan min sinus dan dijumlahkan untuk mendapatkan sinyal QAM-nya. *Carrier* dihasilkan dengan menggunakan *look-up table* yang sudah terdapat pada ROM dari TMS320C542 pada alamat FE00h. Simulasi saluran dilakukan pada keluaran dari modulator kemudian hasilnya diubah ke bentuk analog oleh

DAC dan dikirimkan ke penerima. Simulasi dilakukan dengan menggunakan filter lolos rendah *butterworth* orde 4 dengan frekuensi *cut-off* 2500 Hz dan *random noise generator* dengan tegangan rata-rata 0,3 V (ekivalen dengan kabel 18 kilofeet 26 AWG).



Gambar 8 Diagram Blok Penerima

Diagram blok penerima ada pada gambar 8, sinyal yang datang dikonversi ke bentuk digital oleh ADC. Kemudian sinyal dilewatkan pada filter *passband root raised cosine* dan filter *passband Hilbert transform root raised cosine*. Filter ini merupakan pasangan dari filter *baseband root raised cosine* yang ada di pemancar. Sifat *Hilbert transform* diperlukan untuk membentuk bagian imajiner dari sinyal yang diterima. Keluaran dari filter diteruskan ke *equalizer* untuk secara lebih lanjut menghilangkan efek *intersymbol interference*. Keluaran dari *equalizer* didemodulasi untuk mendapatkan kembali komponen *inphase* dan *quadrature* yang dikirim.

Hasil demodulasi tidak tepat sama dengan nilai komponen *inphase* dan *quadrature* yang seharusnya dari simbol yang dikirim karena masih adanya pengaruh *noise* dan *intersymbol interference* yang masuk. Selain itu, *carrier* pada penerima yang tidak persis sinkron dengan *carrier* dari sinyal yang diterima dapat menyebabkan

pelemahan dan *cross talk* antara komponen *inphase* dan *quadrature*. Untuk mengatasi hal ini digunakan *decision unit*, yaitu unit yang melakukan perhitungan terhadap hasil demodulasi untuk menentukan titik mana yang dikirim. Perhitungan yang dilakukan adalah mencari jarak Euclidean antara titik yang diterima dengan titik-titik konstelasi ideal. Jarak Euclidean adalah jarak dalam konteks ruang yang dihitung dengan rumus sebagai berikut: Jika titik yang diterima memiliki koordinat $(I'(t), Q'(t))$ dan suatu titik ideal memiliki koordinat (x, y) , maka jarak Euclidean-nya adalah:

$$E = \sqrt{\{y - Q'(t)\}^2 + \{x - I'(t)\}^2} \quad (46)$$

Simbol yang dipilih adalah simbol yang jaraknya paling kecil dengan titik yang diterima. Keluaran dari *decision unit* masuk ke *inverse mapper* untuk mengembalikan komponen *inphase* dan *quadrature* yang diterima menjadi bit-bit.

Carrier dari penerima harus disinkronkan dengan *carrier* dari sinyal yang diterima. Proses sinkronisasi *carrier* dilakukan oleh unit *carrier recovery*. Unit *carrier recovery* menggunakan hasil *decision* dan hasil demodulasi untuk mencari *error* fase. *Carrier* dihasilkan dengan menggunakan *look-up table* seperti pada pemancar.

Pelemahan pada sistem dikompensasi oleh *Automatic Gain Control* (AGC). AGC menggunakan hasil demodulasi dan *decision* untuk mencari *gain* yang sesuai. Proses pencarian *gain* melibatkan proses pembagian yang tidak dapat dilakukan secara langsung oleh DSP. Untuk itu, proses ini dilakukan dengan menggunakan *look-up table*. Pada sistem juga terdapat unit *clock recovery* yang bekerja untuk mendeteksi pertengahan dari setiap simbol.

Proses *clock recovery* dilakukan dengan menghitung jumlah dari kuadrat komponen *inphase* dan *quadrature*. Hasil dari penjumlahan tersebut dilewatkan pada sebuah *Band Pass Filter* dengan *bandwidth* yang sangat sempit (*resonator*). Frekuensi *center* dari *resonator* tersebut dibuat sebesar frekuensi simbol yang digunakan. Keluaran dari resonator adalah sebuah sinyal sinus yang memiliki frekuensi sama dengan frekuensi simbol. Sinyal sinus tersebut dilewatkan pada sebuah unit yang mendeteksi pertengahan dari sinyal sinus tersebut.

Pengujian terhadap sistem dilakukan per modul dan secara keseluruhan dengan menggunakan simulasi saluran. Pengujian per modul menunjukkan bahwa modul-modul yang dibuat telah bekerja dengan baik kecuali modul *clock recovery*. Penggunaan metode

clock recovery yang lebih baik akan dapat mengatasi hal ini. Dari hasil pengujian keseluruhan unit didapatkan *error rate* sistem sebesar 1,6825 %.

SPEKIFIKASI SISTEM

- Menggunakan modul DSKplus TMS320C54x dengan prosesor TMS320C542.
- Tegangan catu yang digunakan oleh sistem ini adalah tegangan DC +5 V untuk DSKplus.
- Menggunakan komputer PC/AT dengan sistem operasi Windows 95 untuk berkomunikasi dengan DSKplus.
- Modulasi yang digunakan adalah 8-QAM (*Quadrature Amplitude Modulation*).
- Frekuensi Pencuplikan = 9259,259 Hz.
- Kecepatan Simbol = 2314,815 baud dan kecepatan transmisi data = 6944,445 bps.
- Frekuensi *Carrier* = 1736,11 Hz
- Interferensi antar simbol diatasi dengan filter akar kosinus dinaikkan yang berpasangan membentuk filter kosinus dinaikkan.
- *Equalizer* digunakan untuk mengkompensasi karakteristik dari saluran transmisi.
- Filter adaptif untuk *equalizer* menggunakan algoritma *Least Mean Square (LMS)*.
- Saluran transmisi disimulasikan dengan filter *butterworth* orde empat pada frekuensi *cutt-off* 2500 Hz dan *random noise generator* dengan tegangan rata-rata 0,3 V (ekivalen dengan kabel 18 kilofeet 26 AWG).

KESIMPULAN DAN SARAN

Kesimpulan

1. Hasil pengujian menunjukkan bahwa unit *mapper*, modulator, AGC, demodulator, *decision unit*, dan *inverse mapper* telah bekerja dengan baik.
2. Pengujian unit *carrier recovery* menunjukkan bahwa unit ini dapat mengatasi perbedaan fase dan frekuensi antara *carrier* lokal dengan *carrier* dari sinyal yang diterima.
3. Hasil simulasi menunjukkan bahwa sistem relatif cukup kebal terhadap *noise* (*error rate* = 1,6825 %).

4. Filter akar kosinus dinaikkan *baseband* dan filter akar kosinus dinaikkan *passband* secara kolektif akan menghasilkan tanggapan frekuensi filter kosinus dinaikkan yang akan meminimalkan interferensi antar simbol yang terjadi dan sebagai pasangan filter dicocokkan yang akan meredam noise dalam rentang frekuensi tertentu.
5. *Hilbert Transform* memberikan pergeseran fase sekitar 90^0 untuk membangkitkan bagian imajiner dari sinyal modulasi yang dikirim.
6. Akibat tidak tepatnya pergeseran fase 90^0 oleh Hilbert Transform akan menyebabkan output dari *phase detector* berosilasi disekitar nilai kestabilannya sehingga diperlukan *loop filter*
7. *Loop filter* akan meredam komponen frekuensi tinggi dari *phase detector*, sehingga error fase tidak berosilasi ketika mencapai kestabilan.
8. *Resonator* akan mengekstrak frekuensi simbol dari penjumlahan kuadrat *inphase* dan *quadrature* demodulasi yang digunakan untuk sinkronisasi *clock*.
9. Nilai konstanta μ menentukan kecepatan konvergensi dari kuadrat error *adaptif equalizer*, semakin besar μ maka kecepatan konvergensi makin besar namun dapat menyebabkan *overshoot* dan osilasi disekitar titik kestabilannya.

Saran

Saran untuk menambah kinerja dari sistem dan untuk pengembangan lebih lanjut adalah :

1. Penggunaan metode *clock recovery* yang lebih baik.
2. *Error rate* dari sistem dapat diperkecil dan *symbol rate* dapat dipercepat dengan menggunakan *forward error correction* seperti *Trellis Coding*.
3. Kecepatan pengiriman data dapat ditingkatkan dengan memperbanyak jumlah simbol.
4. Penggunaan *frame* untuk pengiriman data yang panjangnya lebih dari jumlah bit per simbol.
5. Menggunakan prosesor sinyal digital tipe *floating point* untuk memperkecil kesalahan kuantisasi koefisien
6. Melakukan analisa efek transien dari filter dan menggunakan algoritma filter adaptif yang lebih baik seperti *Recursive Least-Squares Algorithm (RLS)*.

DAFTAR PUSAKA

- Anonim. (1996). *TLC320AC01 Single-supply Analog Interface Circuit Data Manual*. Texas Instruments, Texas.
- Anonim. (1996). *TMS320C54x, TMS320LC54x, TMS320VC54x Fixed-Point Digital Signal Processors*. Texas Instruments, Texas.
- Anonim. (1996). *TMS320C54x DSKplus Users's Guide*. Texas Instruments, Texas.
- Anonim. (1996). *TMS320C54x DSP Reference Set Volume 4 : Application Guide*. Texas Instruments, Texas.
- Anonim. (1996). *TMS320C54x DSP Reference Set Volume 3 : Algebraic Instruction Set*. Texas Instruments, Texas.
- Anonim (1996). *TMS320C54x DSP Reference Set Volume 1 : CPU And Peripherals*. Texas Instruments, Texas.
- Buehler, C. J. et al. (1997). *V.34 Transmitter and Receiver Implementation on the TMS320C50 DSP*. Texas Instruments, Texas.
- Defatta, David J. ; Lucas, Joseph G. ; Hodgkiss, William S. (1995). *Digital Signal Processing : A System Design Approach*. John Wiley & Son (SEA) PTE LTD.
- Hayes, Manson H. (1999) *Digital Signal Processing. Schaum's Outline Series*. McGraw-Hill.
- Ifeachor, Emmanuel C. ; Jervis, Barrie W. (1993). *Digital Signal Processing : A Practical Approach*. Addison-Wesley Publishing Company.
- Jeong-Eun Lee; Seung H. Choi; Beomsup Kim. (1997). Baseband Clock Recovery Algorithm for $\pi/4$ -QPSK Modulated Signals. *IEEE Vehicular Technology Conference*, pp. 1729-1733.
- Massey, Tim; Iyer, Ramesh. (1997). *DSP Solutions for Telephony and Data/Facsimile Modems*. Texas Instruments, Texas.
- Proakis, J.G. ; Manolakis, D.G. (1996). *Digital Signal Processing : Principles, Algorithms, and Applications, Third Edition*. Prentice-Hall, New Jersey.
- Proakis, J.G et all. (1992). *Advanced Digital Signal Processing*. Macmillan Publishing Company, New York.
- Schwartz, Mischa. (1986). *Information Transmission, Modulation, And Noise, Third Edition*. McGraw-Hill, Inc.

- Shenoi, Kishan. (1995). *Digital Signal Processing In Telecommunications*. Prentice-Hall, New Jersey.
- Smith, Steven W. (1999). *The Scientist and Engineer's Guide to Digital Signal Processing, Second Edition*. California Technical Publishing, San Diego, California.
- Ziemer, Rodger E. ; Tranter, William H. ; Fannin, D.Ronald. (1990). *Signal and Systems : Continuous and Discrete, Second Edition*. Macmillan Publishing Company, New York.