



(21)申請案號：096143879

(22)申請日：中華民國 96 (2007) 年 11 月 20 日

(51)Int. Cl. : H04L27/26 (2006.01)

H03D3/00 (2006.01)

(30)優先權：2006/11/20	美國	60/866,519
2006/11/20	美國	60/866,527
2006/11/20	美國	60/866,532
2007/03/21	美國	60/907,111
2007/03/22	美國	60/907,126
2007/04/18	美國	60/907,819
2007/05/31	美國	60/940,998
2007/11/13	美國	11/938,848

(71)申請人：美國博通公司(美國) BROADCOM CORPORATION (US)

美國

(72)發明人：紮克 李維 ZAK LEVI (US)；伊利亞赫 舒斯特曼 ELIAHU SHUSTERMAN (US)

(74)代理人：莊志強

(56)參考文獻：

US 6831899B1	US 6985437B1
US 2002/0085654A1	US 2004/0037366A1
US 2004/0066857A1	US 2004/0224715A1
US 2005/0008092A1	US 2005/0122895A1
US 2006/0056554A1	US 2006/0068708A1
WO 2005/117322A1	

Parot, R. et al., "Resolving and correcting gain and phase mismatch in transmitters and receivers for wideband OFDM systems",  
 CONFERENCE RECORD OF THE 36TH. ASILOMAR CONFERENCE ON SIGNALS,  
 SYSTEMS, & COMPUTERS. PACIFIC GROOVE, CA, NOV. 3 - 6, 2002.

審查人員：蕭明椿

申請專利範圍項數：8 項 圖式數：10 共 0 頁

(54)名稱

補償資料信號的第一分量和第二分量間失衡的方法及系統

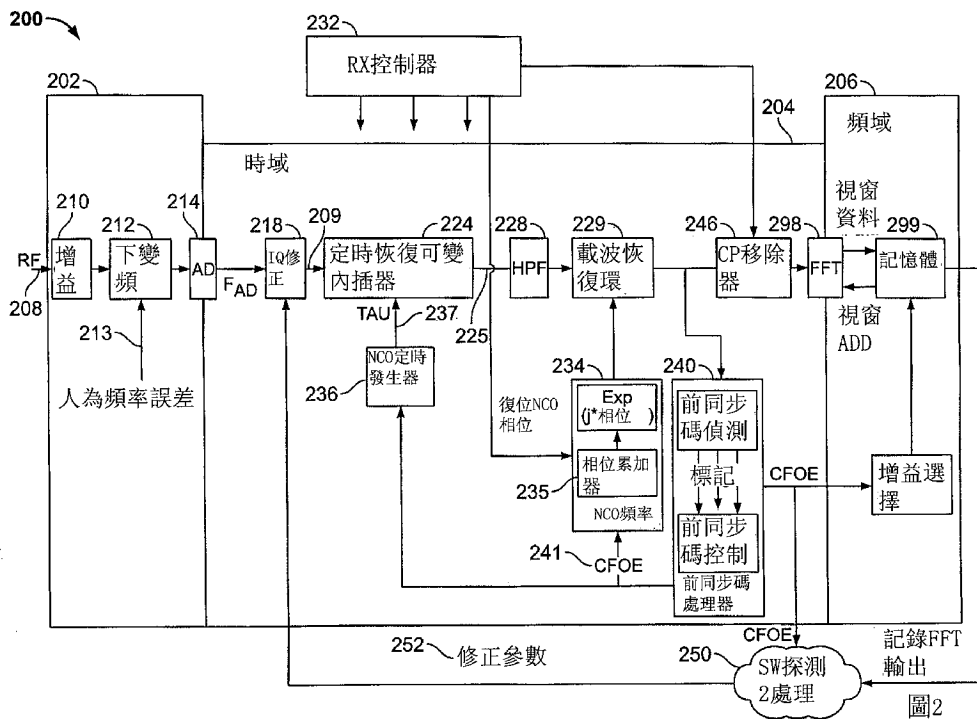
APPARATUS AND METHODS FOR COMPENSATING FOR SIGNAL IMBALANCE IN A RECEIVER

(57)摘要

本發明涉及用於補償 I/Q 失衡的裝置、方法和系統，可包括補償資料信號的第一分量和該資料信號的第二分量之間的失衡。通過具有頻率誤差的載波信號可調製該資料信號。可使用至少一參數表徵所述第一分量。該方法可包括接收資料和載波信號；選擇參數值，以便降低負頻率形式的頻率能量；和基於所述值，修改至少一所述分量。

Apparatus, methods and systems for compensating for an I/Q imbalance may include compensating for an imbalance between a first component of a data signal and a second component of the data signal. The

data signal may be modulated by a carrier signal having a frequency error. The first component may be characterized by at least one parameter. The method may include receiving the data and carrier signals; selecting a value for the parameter such that the frequency domain energy at negative frequencies is reduced; and modifying at least one of the components based on the value.



- 200 . . . 接收器
- 202 . . . 射頻(“RF”)處理模組
- 204 . . . 時域處理模組
- 206 . . . 頻域處理模組
- 208 . . . RF 信號
- 209 . . . 信號
- 210 . . . 增益
- 213 . . . 人為頻率誤差
- 214 . . . 模數轉換器
- 218 . . . 失衡補償模組
- 224 . . . 可變速率內插器
- 225 . . . 信號
- 228 . . . 高通濾波器(“HPF”)
- 229 . . . 載波恢復環
- 232 . . . 接收器控制器
- 234 . . . NCO 頻率發生器
- 236 . . . 數控振盪器(“NCO”)定時發生器
- 237 . . . 定時信號
- 240 . . . 前同步碼處理器
- 241 . . . 載波頻率偏移估計(“CFOE”)
- 246 . . . CP 移除器
- 250 . . . 探測 2 軟體處理程式

圖2

- 252 . . . 修正參數
- 298 . . . 快速傅立葉  
變換模組
- 299 . . . 記憶體

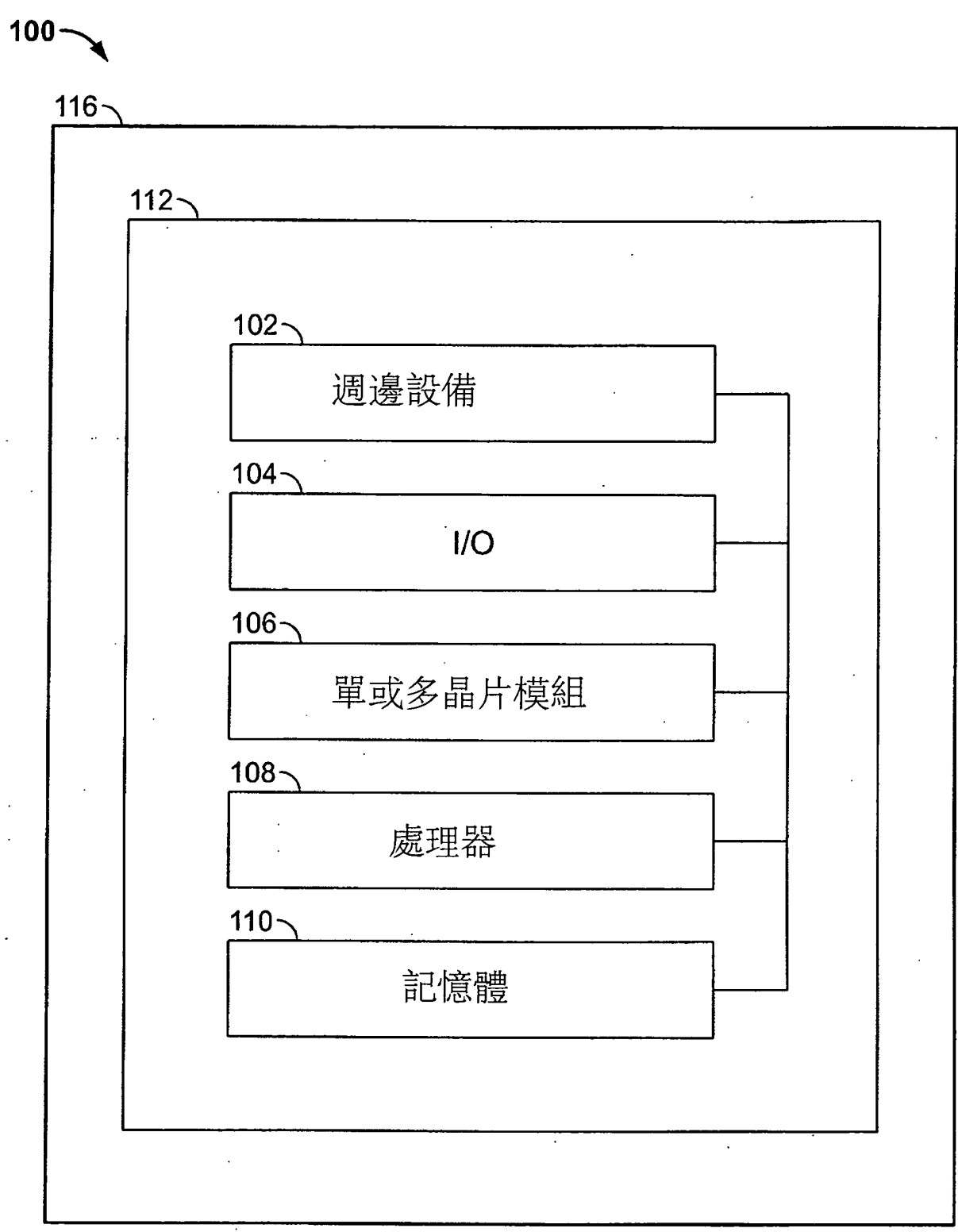


圖1

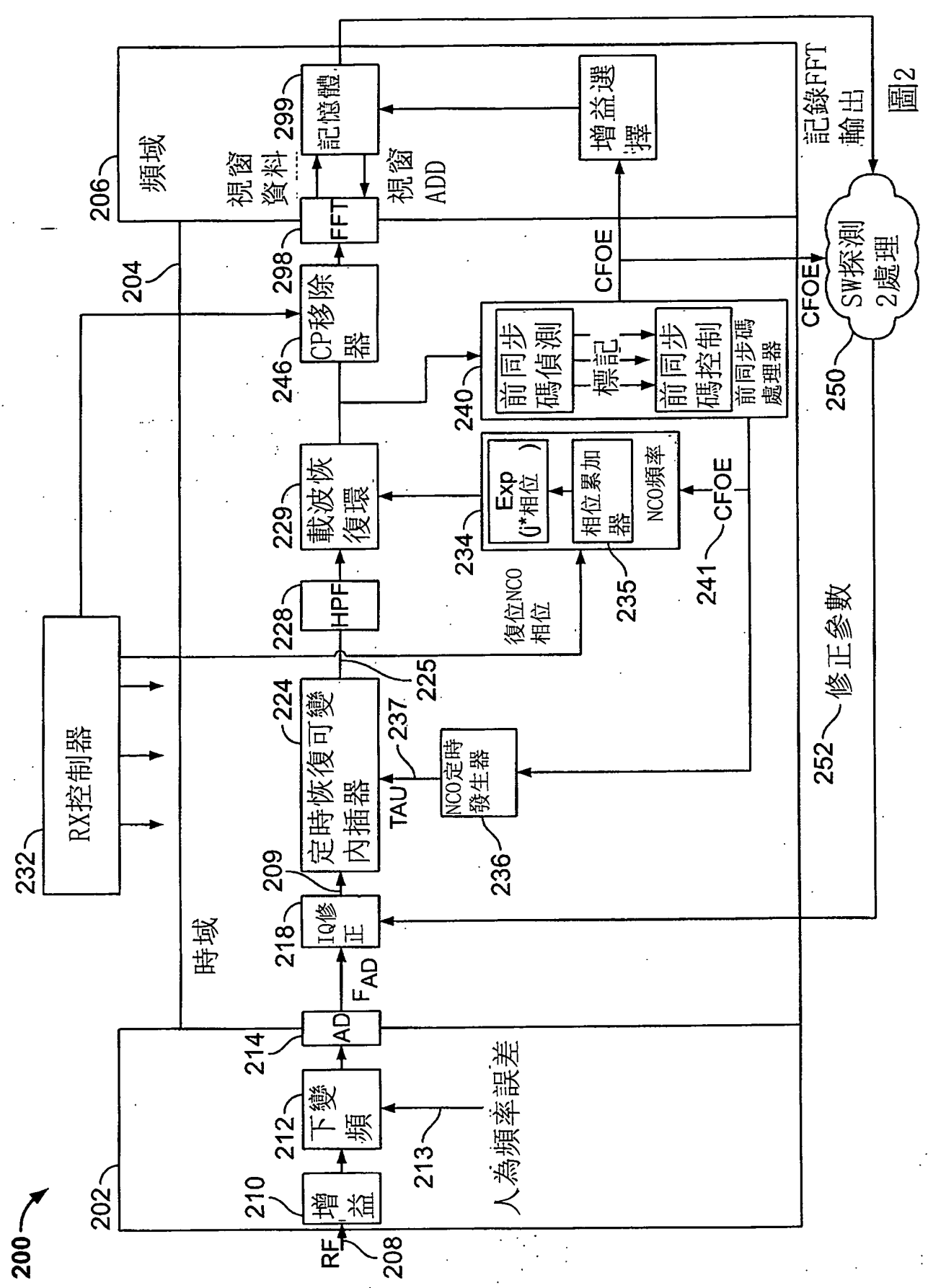


圖2

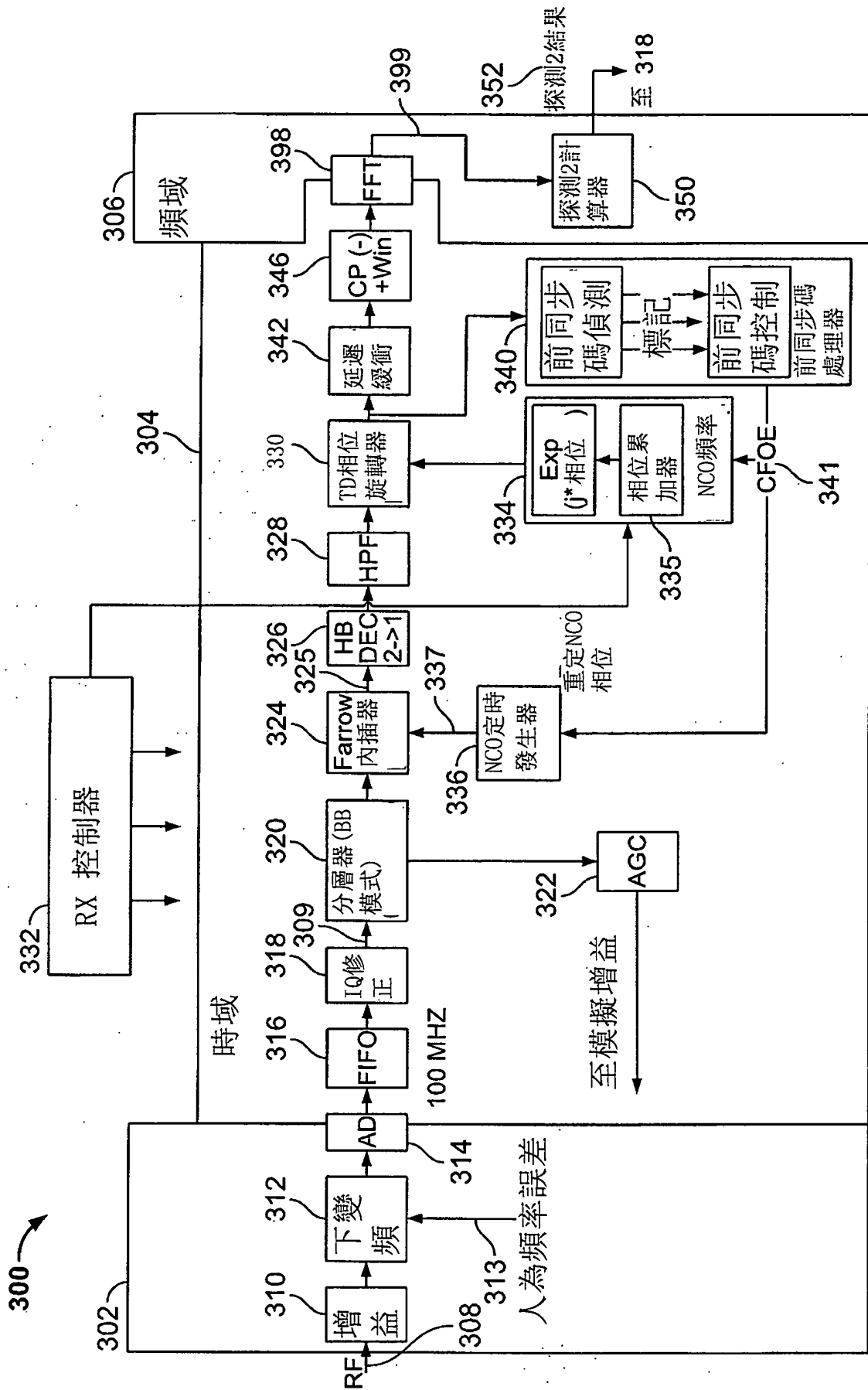


圖3

400

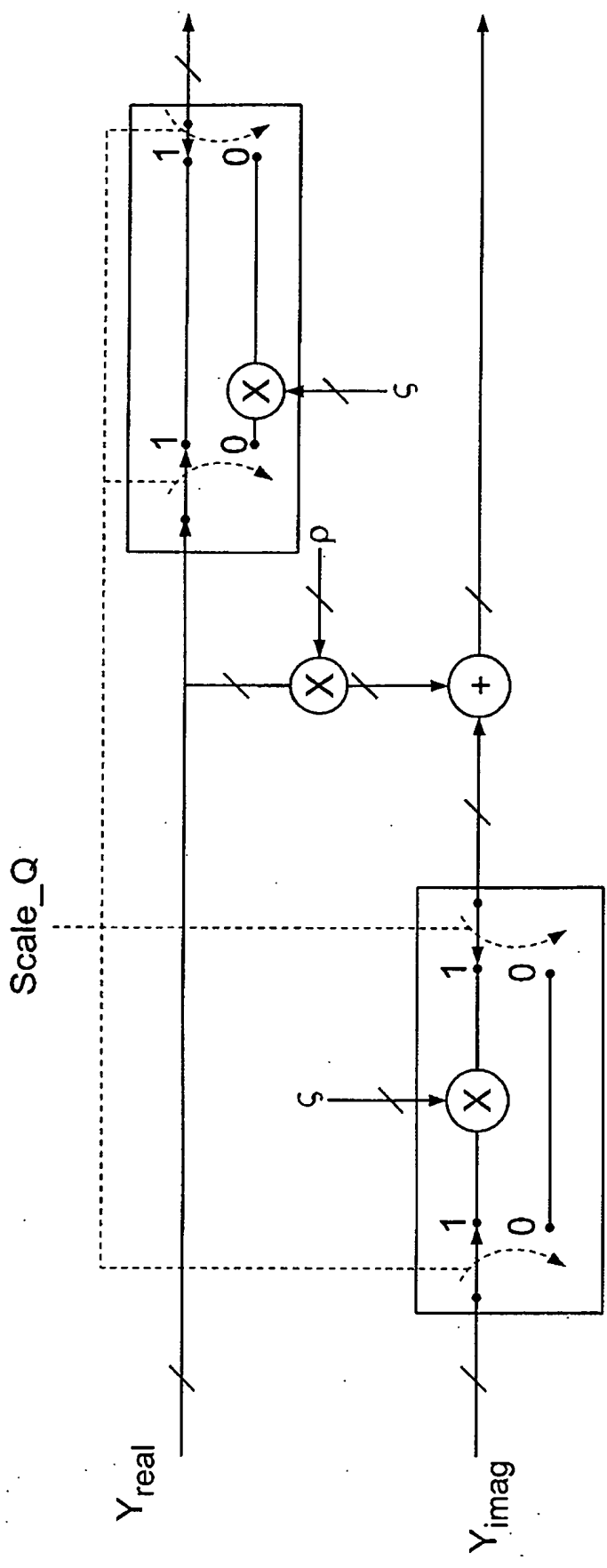


圖4



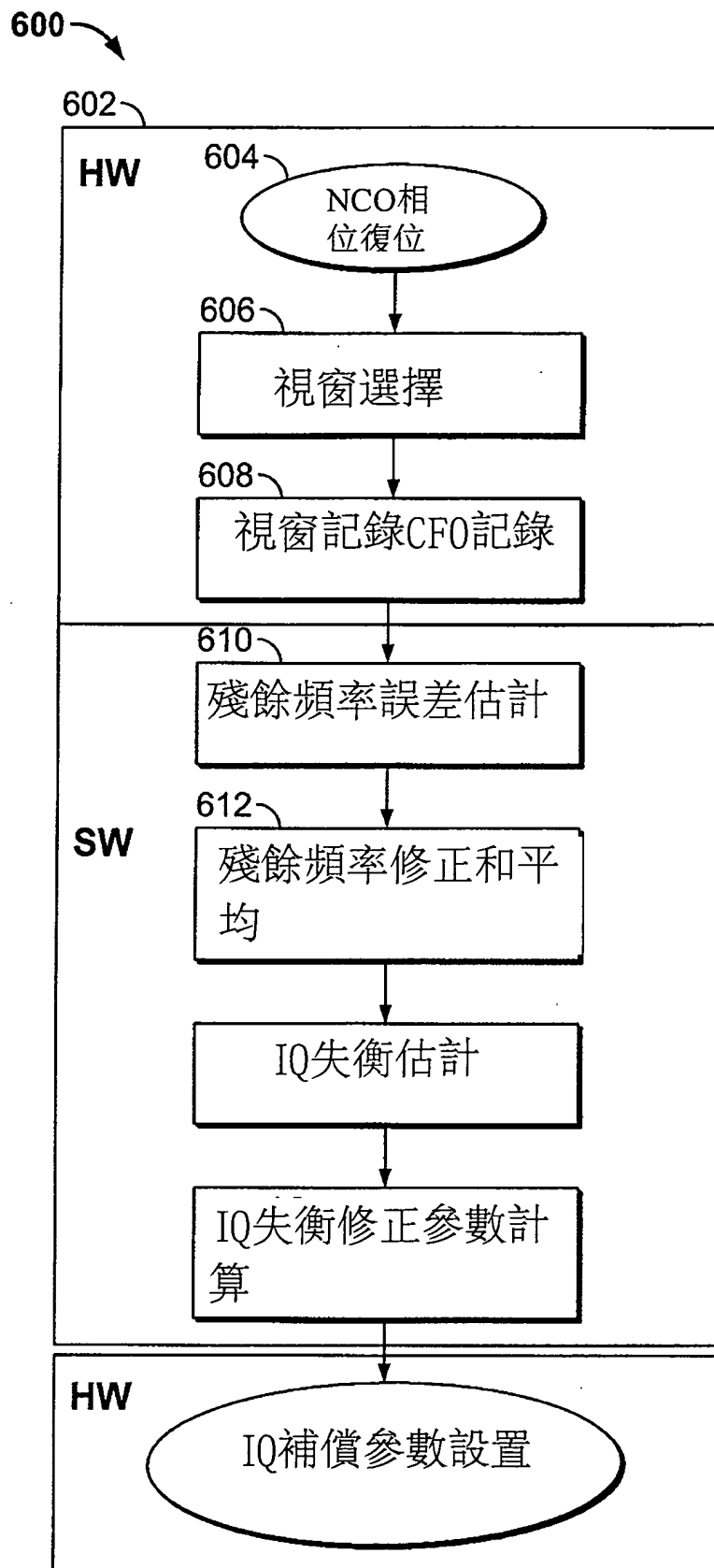
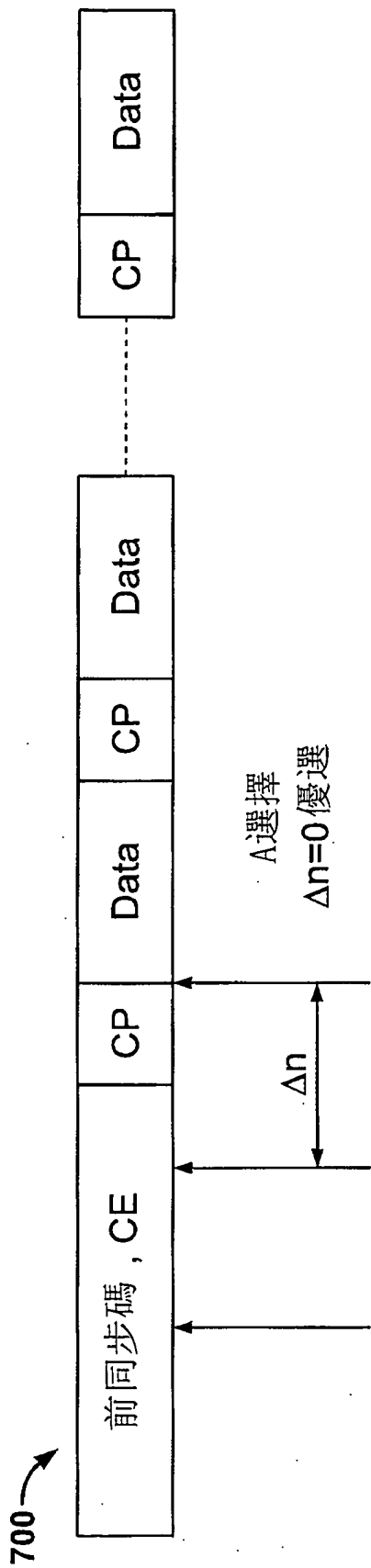


圖6



精確的頻率 NCO復位  
修正

FFT視窗的  
開始

NCO復位

圖7

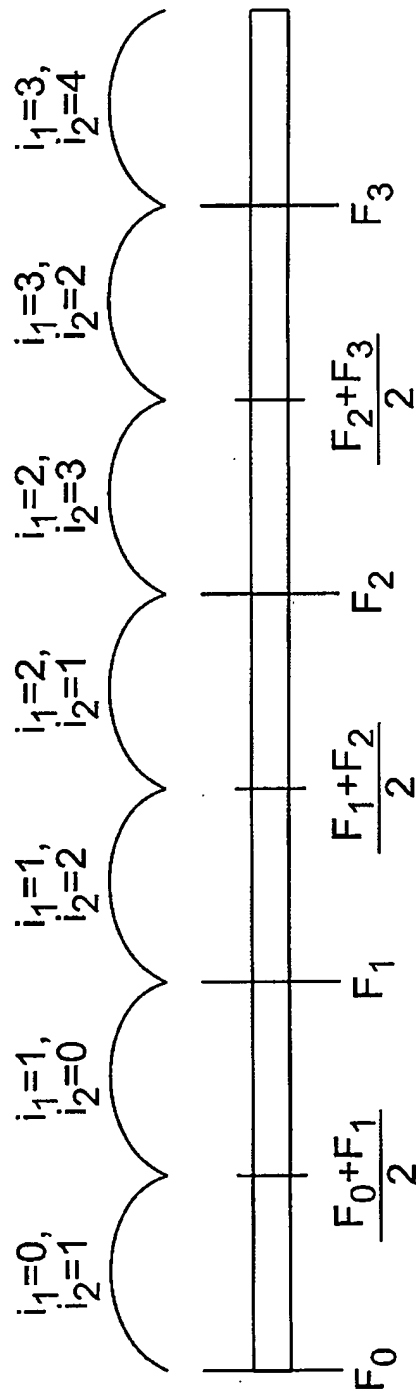


圖8



$L = N_{\text{sym}} - 1$

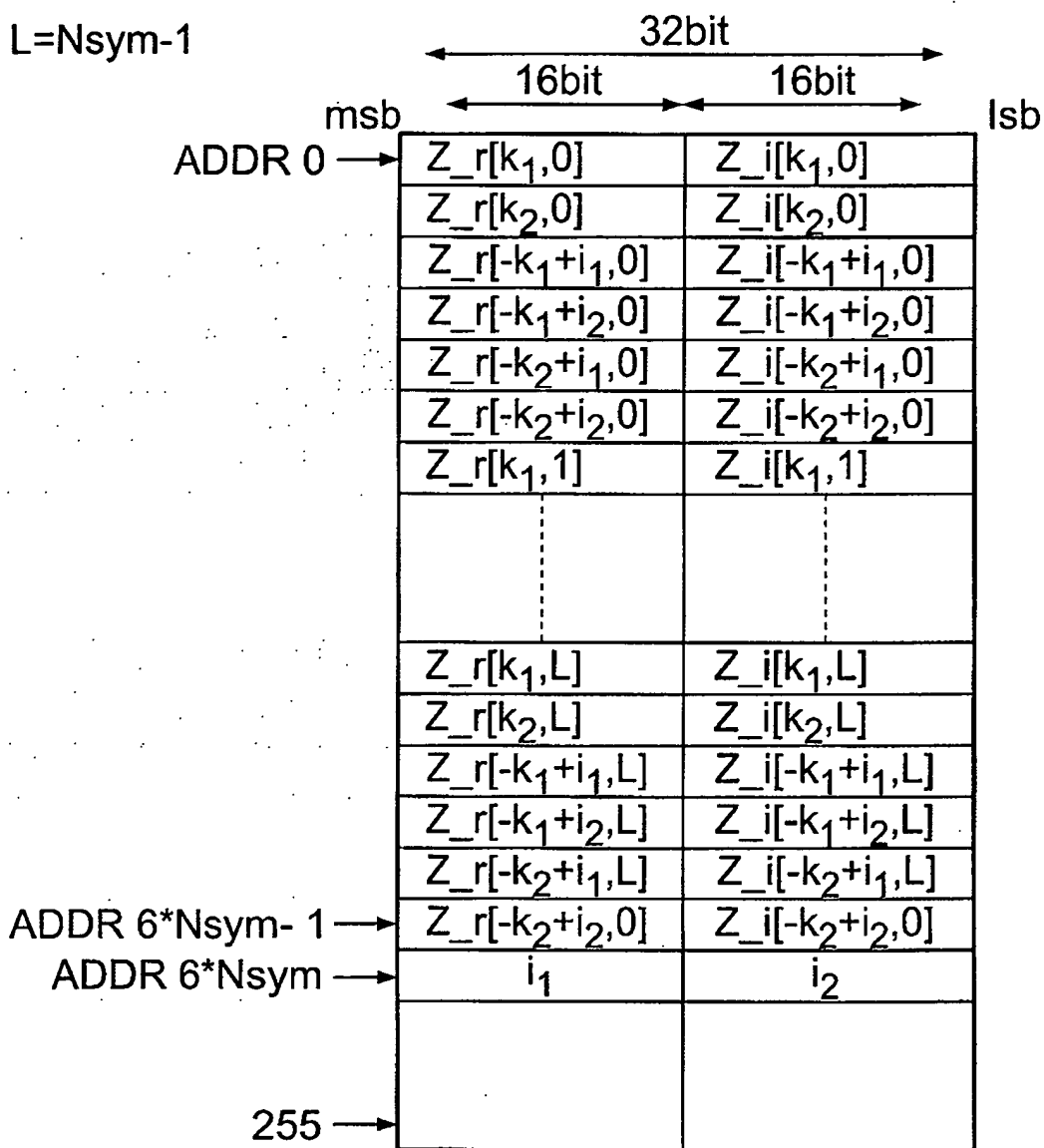


圖 10

# 發明專利說明書

(本說明書格式、順序，請勿任意更動)

## 【發明名稱】(中文/英文)

補償資料信號的第一分量和第二分量間失衡的方法及系統 /  
APPARATUS AND METHODS FOR COMPENSATING FOR  
SIGNAL IMBALANCE IN A RECEIVER

## 【技術領域】

本發明涉及資訊網路，更具體地說，涉及通過通信線路例如同軸電纜，發送資訊例如多媒體資訊，以形成通信網路。

## 【先前技術】

許多建築物，包括家庭，都有基於同軸電纜的網路。

同軸電纜多媒體聯盟 (“MoCA<sup>TM</sup>”) 在其網站 ([www.mocalliance.org](http://www.mocalliance.org)) 上提供了通過同軸電纜聯播數位視頻和娛樂資訊的規範(即，其在 MoCA 內所使用，並在此全文引用)的一例子。該規範分發給公開註冊的用戶。

基於 MoCA 和其他規範的技術、以及相關的技術(“現有的技術”)常常使用同軸電纜上可使用的並且未使用的帶寬。例如，在美國，已超過 70% 的家庭安裝了同軸電纜。一些家庭在一個或多個娛樂消耗區域已有現成的同軸電纜，例如家庭活動室、視聽室、和主臥室。現有的技術使得屋主可使用已安裝的同軸電纜，作為網路系統，並發送具有高服務質量(QoS)的娛樂和資訊程式。

現有的技術可提供高速度(270mbps)、高 QoS、以及最高級別的封包級加密和遮罩的有線連接兩者所固有的安全性。同軸電纜是設計用於發送高帶寬視頻。目前，同軸電纜通常用於安全傳送數百萬美元的按次計費及優質視頻內容。基於現有技術的網路可用作多個無線接入點的骨幹網，從而擴展無線服務在建築物內的覆蓋範圍。

現有的技術通過現有的同軸電纜提供了到達目前安裝在家庭中的視頻設備所在地的吞吐量，而不影響線纜中的其他服務信號。現有的技術為數位娛樂提供鏈路，並與別的有線或無線網路保持一致，從而將娛樂流量擴展到整個建築物。

現有技術與接入技術協同工作，如非對稱數位用戶線路(“ADSL”)、甚高速率數位用戶線(“VDSL”)和光纖到戶(“FTTH”)，進而提供一般通過雙絞線或光纖進入該建築物的信號，對於 ADSL 而言，其操作頻帶從幾十萬赫茲到 8.5M 赫茲，而對於 VDSL 是到 12M 赫茲。當服務通過任何數位用戶線(“xDSL”)或 FTTH 到達該建築物時，這些服務可通過現有技術和同軸電纜轉發到視頻設備。有線電視運營商可通過線纜為該建築物提供諸如視頻、音頻和因特網接入等有線功能，並利用在該建築物中運行的同軸電纜到達該建築物中的各個有線服務使用設備。一般地，現有技術的功能與有線功能並行，不過是在不同頻率上。

建築物中的同軸電纜設施一般包括同軸電纜、分配器和出口。分配器一般有一個輸入和兩個或多個輸出，用於前向(輸入到輸出)或後向(輸出到輸入)傳送信號，並將來自不同分配器的輸出相隔離，從而防止信號從一個同軸電纜出口流向另一個。進行隔離是非常有用的，因為它可以 a)減少來自其他設備的干擾和 b)最大化從入口點(“POE”)到出口的功率傳輸以獲得最佳 TV 接收。

現有技術的部件(elements)被專門用於通過隔離器進行後向傳播(“插入”)以及從輸出到輸出的後向傳播(“隔離”)。通過特定“隔離跳”和多個“插入跳”可從建築物中的一個出口到達另一個。一般隔離跳的衰減為 5 到 40dB，而每個插入跳大約將衰減 3dB。MoCA™ 技術具有 55dB 盈餘的動態範圍，同時支援 200Mbps 的吞吐量。因此，MoCA™ 技術可通過大量隔離器高效運行。

管理網路策略，諸如 MoCA™ 技術，專門設計用於支援沒有資料包丟失的流視頻，進而提供出口間的優質視頻。

當網路連接設備從網路接收到資料信號時，該信號在下變頻為設備基帶頻率時，常常分解為同相(“I”)和正交(“Q”)分量；當該 I 和 Q 分量再組合，以進行資料解密時，它們常常是振幅、相位或兩者失衡的。再平衡 I 和 Q 分量將涉及計算基於載波頻率的頻域簽名以及 I 和 Q 分量的補償因數。在載波頻率不確定的情況下，通過使用數位計算方法，很難分析接收到的信號的頻域簽名。因此，期望提供在載波頻率不確定的情況下使用數位計算方法的補償信號的系統和方法。

### 【發明內容】

一種用於通信網路中在節點補償 I/Q 失衡的系統和/或方法，在至少一張附圖中進行了描述，並在權利要求中進行了完整的說明。

根據本發明的一個方面，提供一種用於補償資料信號的第一分量和第二分量之間失衡的方法，所述資料信號具有正值的頻率，並通過具有頻率誤差的載波信號對其進行調製，包括以下步驟：接收資料和載波信號；選擇時域參數值，以使與負值頻率相應的頻域能量最小化；和基於所述值，在時域中修改至少一個所述分量。

優選的，所述的方法進一步包括：估計與所述載波信號相應的載波信號頻率；估計殘餘載波頻率誤差；和基於所述殘餘載波頻率誤差，修改所述載波頻率的估計。

優選的，所述修改包括：為所述分量確定與信號振幅乘法器相應的第一值；和為所述分量確定與相位偏移相應的第二值。

優選的，所述選擇包括：為第一信號音、第二信號音、以及所述載波信號中的每一個記錄第一頻域參數和第二頻域參數。

優選的，對於所述第一信號音、所述第二信號音、以及所述載波信號，其中至少有一個其所述第一和第二頻域參數分別對應於離散值頻譜中的第一和第二窗口；所述第一窗口順次地鄰近所述

第二窗口。

優選的，所述的方法進一步包括：發送取決於所述第一和第二頻域參數的輸出信號。

優選的，所述接收包括：接收正交頻分複用(OFDM)符號；和在所述選擇中，所述值對應於所述 OFDM 符號；和對於所述 OFDM 符號，執行所述修改。

優選的，所述接收包括：接收多個正交頻分複用(OFDM)符號；對每一個所述 OFDM 符號，執行所述選擇；和對每一個所述 OFDM 符號，執行所述修改。

優選的，所述第一和第二分量，其中一個是同相分量；另一個是正交分量；所述失衡是同相/正交失衡。

根據本發明的一個方面，提供一種可記錄與接收到的信號的頻率分量相應的複數參數的系統，所述信號攜帶有至少一個正交頻分複用(OFDM)符號，所述參數符合：具有頻率誤差的載波頻率；第一信號音；和第二信號音。

優選的，所述系統還用於估計與所述載波頻率、所述第一信號音、以及所述第二信號音中的每一個相應的頻率視窗。

優選的，當具有與所述載波頻率、所述第一信號音、以及所述第二信號音中的每一個相應的第一和第二複數參數時，所述系統進一步用於：估計與所述載波頻率、所述第一信號音、以及所述第二信號音中的每一個相應的第二頻率視窗；為所述載波頻率、第一信號音、以及第二信號音記錄所述第一和第二複數參數。

優選的，所述系統進一步用於根據所述載波頻率、所述第一信號音、以及所述第二信號音中的每一個的估計值，為所述載波頻率、所述第一信號音、以及所述第二信號音，記錄所述複數參數；所有的所述估計值相應于單一 OFDM 符號。

優選的，當所述接收到的信號攜帶多個 OFDM 符號時，所述系統進一步可對於每一個所述 OFDM 符號，估計所述載波頻率、

所述第一信號音、和所述第二信號音中的每一個的值。

根據本發明的一個方面，提供一種用於補償第一信號和第二信號之間失衡的系統，該系統包括：第一模組，用於根據所述第一和第二信號記錄第一頻域參數、第二頻域參數；所述第一頻域參數和第二頻域參數相應於所述第一信號音、第二信號音、以及載波頻率中的每一個；和第二模組，用於根據所述第一和第二頻域參數，計算至少一個時域補償參數。

優選的，所述系統進一步包括：第三模組，用於根據所述時域補償參數，輸出經補償的信號。

優選的，所述至少一個時域補償參數包括三個時域補償參數。

優選的，所述載波頻率是接收器載波頻率；所述第一和第二信號接收自發送器，所述發送器可使用發送器載波頻率發送所述信號；和由於載波頻率偏移，所述接收器和發送器載波頻率不同。

優選的，當所述第一和第二頻域參數相應於所述第一和第二信號音的其中之一時，所述第一頻域參數對應於離散值頻譜中的第一窗口；所述第二頻域參數對應於離散值頻譜中的第二窗口；所述第一窗口鄰近所述第二窗口。

優選的，所述第一模組根據所述第一點在頻譜中的位置選擇所述第二窗口。

優選的，所述第二窗口在頻譜中的位置是基於所述偏移的。

優選的，所述第二窗口在頻譜中的位置是基於所述偏移，以弧度表示，到  $\pi$  弧度。

優選的，當具有殘餘載波頻率偏移，所述第二模組可：計算殘餘頻率偏移的估計；基於所述估計，計算所述時域補償參數。

優選的，當所述第一模組為按時間順序接收到的多個符號中的每一個記錄所述第一和第二頻域估計參數時，所述第二模組為第一頻率參數計算第一平均值；並為所述第二頻域參數計算第二平均值；所述平均值分別基於與所述符號相應的所述第一和第二頻

域參數。

優選的，所述第一模組是硬體模組；所述第二模組是軟體模組。

優選的，所述第三模組是硬體模組。

優選的，所述第一模組是硬體模組；所述第二模組是硬體模組。

### 【圖式簡單說明】

圖 1 是依據本發明原則的所使用的例證性的單或多晶片設備的結構框圖；

圖 2 是依據本發明原則的接收器的一部分的例證性的結構框圖；

圖 3 是依據本發明原則的接收器的一部分的另一例證性的結構框圖；

圖 4 是依據本發明原則的電路的例證性的結構框圖；

圖 5 是依據本發明原則的電路的另一例證性的結構框圖；

圖 6 是依據本發明原則的例證性的流程圖；

圖 7 是依據本發明原則的所處理的例證性的資料包的簡圖；

圖 8 是依據本發明原則的與信號處理相關的例證性的一部分離散值頻譜的示意圖；

圖 9 是依據本發明原則的另一電路的例證性結構框圖；

圖 10 是依據本發明原則的存儲結構的示意圖。

### 【實施方式】

依據本發明原則，提供用於補償 I/Q 失衡的裝置和方法。該方法包括補償資料信號的第一分量和該資料信號的第二分量之間的失衡。通過具有頻率誤差的載波信號調製該資料信號。可通過至少一個參數表徵該第一分量。該方法可包括接收資料和載波信號；選擇參數值以便降低頻域能量；該頻域能量相應於負的頻率；基於所選擇的值，修改至少一分量。

該裝置包括一電路，其可記錄與接收到的信號的頻率分量相應的信號值。該信號攜帶至少一個正交頻分複用(“OFDM”)符號。

該信號的值可對應於具有頻率誤差的載波頻率、第一信號音(tone) 和第二信號音。

該裝置包括一系統，其用於補償資料信號的第一分量和該資料信號的第二分量。可通過具有頻率誤差的載波信號調製該資料信號。可通過至少一參數表徵該第一分量。該系統包括用於量化信號值的硬體模組，和用於從該硬體模組接收該信號值的軟體模組，其中該信號值相應于資料和載波信號的其中之一。

可通過前述的 MoCA 規範中所列的 MoCA 協定探測 2(probe2) 傳輸，發送第一和第二信號音。

下面將結合附圖 1~8 及附錄 A~E 對本發明作進一步說明。

圖 1 示出了依據本發明的單或多晶片模組 102，其可以是一個或多個積體電路，且位於依據本發明的例證性的資料處理系統 100 中。資料處理系統 100 可包括一個或多個如下的部件：I/O 電路 104、週邊設備 102、處理器 108 和記憶體 110。這些部件可通過系統匯流排或別的互連 112 連接在一起，並設置在終端用戶系統中的電路板上。模組 102 的元件可執行涉及 I/Q 失衡補償的任務。

在一些實施例中，可在 MoCA 探測 2 猝發接收(burst reception) 過程中，執行 I/Q 失衡補償。探測 2 是一 2-信號音信號，其用於接收器中的 I/Q 失衡計算或別的 RF 校準。PHY 層執行視窗(bin) 選擇和記錄，所得的結果將上載到 CPU，用於 I/Q 補償參數的計算。

圖 2 示出例證性的接收器 200 的一部分的結構框圖。接收器 200 可包括射頻(“RF”)處理模組 202、時域處理模組 204 和頻域處理模組 206。在增益 210，接收 RF 信號 208，並進行增益調節。在 212，將信號 208 下變頻為基帶(“BB”)頻率。在 212，將人為頻率誤差(intentional frequency error)213 添加到信號 208。模數轉換器 214 將信號 208 轉換為以模數採樣率採樣的數位信號，並將其發送到失衡補償模組 218。I/Q 失衡補償模組 218 可執行與 I/Q 補

償相關的步驟。I/Q 失衡補償模組 218 輸出信號 209，其對應於等式 1(如下)。

信號 209 發送到可變速率內插器 224，該可變速率內插器 224 以一合適的採樣率，對信號 209 進行再採樣。

可變速率內插器 224 可接收數控振蕩器(“NCO”)定時發生器 236 輸出的定時信號 237。定時信號 237 是基於前同步碼處理器 240 輸出的載波頻率偏移估計(“CFOE”)241。CFOE241 可基於前同步碼處理器 240 估計。內插器 224 輸出信號 225，然後其可通過高通濾波器(“HPF”)228，去除直流(“DC”)信號分量。

載波恢復環 229 可執行對人為頻率誤差 213 的頻率補償。載波恢復環可接收 NCO 頻率發生器 234 的輸入，其中 NCO 頻率發生器 234 由接收器控制器 232 進行控制。NCO 頻率發生器 234 可接收前同步碼處理器 240 輸出的載波頻率偏移估計 241。在 CP 移除器 246，可將迴圈字首從信號 225 去除。

在頻域處理模組 206 中，使用快速傅立葉變換模組 298 將信號 225 轉換成頻域資訊(“FFT 輸出”)，其可存儲在記憶體 299 中，並發送到探測 2 軟體處理程式 250，其可輸出修正參數 252，以返回給 I/Q 失衡補償模組 218。

圖 3 示出例證性的接收器 300 的一部分的結構框圖。接收器 300 可包括射頻(“RF”)處理模組 302、時域處理模組 304 和頻域處理模組 306。在增益 310，接收 RF 信號 308，並調節其增益。在 312，信號 308 下變頻為基帶頻率。在 312，將人為頻率誤差 313 添加到信號 308。模數轉換器 314 將信號 308 為轉換數位信號，並將其發送到 100 MHz FIFO(“先進先出”)緩衝器 316。緩衝器 316 發送信號 308 到 I/Q 失衡補償模組 318。I/Q 失衡補償模組 318 可執行與在此的 I/Q 補償相關的步驟。I/Q 失衡補償模組 318 輸出信號 309，其對應於等式 1(如下)。

信號 309 可發送到基帶模式分層器 320。接收器 300 可包括自

動增益控制器 322，其可基於信號 309 給增益 310 提供反饋。通過分層器 320，信號 309 可發送到 Farrow 內插器 324，其以較低的採樣率，對 100MHz 信號 309 進行再採樣。

Farrow 內插器 324 可接收數控振蕩器(“NCO”)定時發生器 336 輸出的定時信號 337。定時信號 337 是基於前同步碼控制處理器 340 輸出的載波頻率偏移估計 341 的。載波頻率偏移估計 341 是基於經過前同步碼處理器 340 所發送的 TD 相位旋轉器(下面討論)的輸出。在一些實施例中，內插器 324 以 100MHz 輸出信號 325。經過定時恢復環(未示出)，信號 325 可與發送器時鐘(未示出)同步。經過半帶濾波器抽取器(“HB DEC2→1”)236，使用因數 2，將信號 325 向下採樣為 50MHz。然後通過高通濾波器(“HPF”)328，去除信號 325 的直流(“DC”)信號分量。

時域(“TD”)相位旋轉器 330 可執行對人為頻率誤差 313 的頻率補償。TD 相位旋轉器可接收 NCO 頻率發生器 334 的輸入，該 NCO 頻率發生器 334 可由接收器控制器 332 控制。NCO 頻率發生器 334 可接收前同步碼處理器 340 輸出的載波頻率偏移估計 341。信號 325 然後發送到延遲緩衝器 342。在子電路 346，可去除迴圈字首。在一些實施例中，子電路 346 可執行接收器開窗操作，以降低窄帶干擾噪音的損害，該窄帶干擾噪音否則可滲透到相鄰的信號音。

在頻域處理模組 306 中，可使用快速傅立葉變換模組 398 將信號 325 轉換成頻域資訊，其可發送到探測 2 計算器 350，該探測 2 計算器 350 可輸出探測 2 結果 352，並發送到 I/Q 補償模組 318。

一些實施例包括旁路模式，在其中將信號輸入路由到 I/Q 失衡補償模組 318 周邊的輸出。

在一些實施例中，通過數位信號分析和處理可實現 I/Q 補償。在這些實施例中，在探測 2 期間， $\rho$  和 Scale\_Q 為估計的 I/Q 補償參數。

等式 1 示出補償信號的補償實部和虛部，該補償信號是 I/Q 補償模組的輸出(見圖 2)。

### 等式 1

$$\tilde{Y}_{real} = \begin{cases} Bypass = 1 \\ Bypass = 0 \end{cases} \begin{cases} Y_{real} \\ \zeta Y_{real} \\ Y_{real} \end{cases} \quad \tilde{Y}_{imag} = \begin{cases} Bypass = 1 \\ Bypass = 0 \end{cases} \begin{cases} Y_{imag} \\ \tilde{Y}_{imag} = Y_{imag} + \rho Y_{real} \\ \zeta Y_{imag} + \rho Y_{real} \end{cases}$$

圖 4 示出例證性的電路 400，其包括在執行等式 1 所列的補償的設備中。

圖 5 示出例證性的電路 500，其包括在執行等式 1 所列的補償的設備中。附錄 A 列出了等式 1 所列的補償的理論基礎。附錄 B 示出與在此所描述的裝置和方法相聯繫的示範性的失衡和補償度量。

圖 6 示出用於補償 I/Q 失衡的例證性的過程 600。過程 600 涉及硬體(“HW”)和軟體(“SW”)操作。初始的硬體操作 602 的輸出是資料組  $Z(k,m)$ ，其是快速傅立葉變換(“FFT”)在對應於探測 2 的 OFDM 符號  $m$  的視窗  $k$  的輸出。

初始的硬體操作 602 可包括數控振蕩器(“NCO”)相位重定 604。確定 FFT 視窗的第一樣本的相位，該 FFT 視窗是時域單元(“TDU”)頻域補償的結果。為此，在計算出精確的頻率補償後的任意時間，將接收器 TDU 中的相位旋轉器的 NCO 相位重定為零。計算 NCO 重定和 FFT 視窗的第一樣本之間的樣本數量(相位累加數量)，且定義為  $\Delta n$ ，並發送到 SW 程式。最好是零相位累加(即  $\Delta n = 0$ )，因為其可降低 SW 程式的複雜度。對於設置  $\Delta n = 0$ ，一旦 356 點 FFT 視窗的第一樣本傳輸通過 TD 相位旋轉器 330(見圖 3)(因此第一樣本將乘以一整數)，應重定 NCO 相位累加器 335(位於 NCO 頻率發生器 334 中—見圖 3)。

圖 7 示出資料包 700、NCO 重定、 $\Delta n$  和 FFT 視窗的開始。

在一些實施例中，窗口選擇 606(見圖 6)的執行作為浮點計

算，其中  $i_1$  和  $i_2$  是頻率視窗係數，如等式 2 進行計算

### 等式 2

$$i_1 = -\text{round}\left(\frac{N \cdot CFO}{\pi}\right)$$

$$i_2 = i_1 + \text{sign}(CFO) \cdot \text{sign}\left(|i_1| - \left|\frac{N \cdot CFO}{\pi}\right|\right)$$

其中， $CFO/(2\pi)$ 是發射器和接收器之間的經估計的載波頻率偏移， $N$ 是 FFT 視窗的數量(例如 256)。

在一些實施例中，視窗選擇 606(見圖 6)的執行可作為定點計算。在這些實施例中， $CFO$  是 17 比特帶符號整數，其中‘1’= 214。通過與固定閾值比較，計算  $i_1$  和  $i_2$ 。等式 3 給出了在定點表示中，FFT 柵格(grid)的值：

### 等式 3

$$F_k = \text{round}\left(\frac{2\pi k}{N} \cdot 2^{\text{Freq\_bits}-1}\right)$$

其中， $k = [-3, 3]$

其中， $\text{Freq\_bits}$  可設置為 14 或任意合適的數位。通過找到最接近  $2CFO$  的兩個 FFT 窗口，選擇係數  $i_1$  和  $i_2$ 。

圖 8 示出對應於等式 3 的決策區域(由於對稱僅給出絕對值)。

圖 9 示出用於決策區域邊界選擇的例證性的硬體(“HW”)的實施。表 1 示出例證性的邊界值。

表 1

頻率邊界	$\frac{F_0 + F_1}{2}$	$\frac{F_1 + F_2}{2}$	$\frac{F_2 + F_3}{2}$	$F_1$	$F_2$	$F_3$
定點值	804	2413	4012	1608	3217	4825

等式 4 定義為標記(sign)操作。

等式 4

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

在一些實施例中，視窗記錄(步驟 408，見圖 4)涉及在視窗  $k_1$ 、 $-k_1 + i_1$ 、 $-k_1 + i_2$ 、 $k_2$ 、 $-k_2 + i_1$ 、 $-k_2 + i_2$  的 16 比特 FFT 輸出，然後為每一個 L OFDM 符號記錄該輸出。可以理解的是 FFT 的輸出可為任意合適的數量的比特。MoCA FFT 中的位址對應於表 2 所列的窗口。

表 2

$ADDR: k_1$	$Z[k_1, m]$	$k_1 \in [146, 186]$
$ADDR: k_2$	$Z[k_2, m]$	$k_2 \in [217, 249]$
$ADDR: 256 - k_1 + i_1$	$Z[-k_1 + i_1, m]$	$-k_1 + i_1 = 256 - k_1 + i_1 \in [67, 113]$
$ADDR: 256 - k_1 + i_2$	$Z[-k_1 + i_2, m]$	$-k_1 + i_2 = 256 - k_1 + i_2 \in [66, 112]$
$ADDR: 256 - k_2 + i_1$	$Z[-k_2 + i_1, m]$	$-k_2 + i_1 = 256 - k_2 + i_1 \in [4, 36]$
$ADDR: 256 - k_2 + i_2$	$Z[-k_2 + i_2, m]$	$-k_2 + i_2 = 256 - k_2 + i_2 \in [3, 37]$

圖 10 示出在探測 2 猝發末端處的例證性的存儲圖。

在步驟 608 記錄 CFO 估計(如圖 6 所示)。該估計可為 17-比特估計。

在一些實施例中，可通過軟體模組執行殘留頻率誤差估計 610\_見圖 6。在一些實施例中，可通過硬體模組執行殘留頻率誤差估計。在一些實施例中，可如等式 5 所示計算殘留頻率誤差估計。

等式 5

$$\hat{\epsilon} = \frac{\text{angle} \left( \sum_{m=0}^{L-2} Z[k_i, m] Z^*[k_i, m+1] \right)}{2\pi(N + N_{CP})}$$

其中，

$$k_i = \begin{cases} k_1 & SNR_{k_1} > SNR_{k_2} \\ k_2 & SNR_{k_2} > SNR_{k_1} \end{cases}$$

在一些實施例，可依據等式 6，計算殘餘頻率誤差補償和時間平均，其取決於  $\hat{\varepsilon}$ ，並且附錄 A 列出其導數。

### 等式 6

$$\begin{aligned}\bar{Z}_{k_1} &= \sum_{m=0}^{L-1} Z[k_1, m] e^{-j2\pi\hat{\varepsilon}(N+N_{CP})m} & \bar{Z}_{k_2} &= \sum_{m=0}^{L-1} Z[k_2, m] e^{-j2\pi\hat{\varepsilon}(N+N_{CP})m} \\ \bar{Z}_{-k_1+i_1} &= \sum_{m=0}^{L-1} Z[-k_1+i_1, m] e^{-j(2CFO+2\pi\hat{\varepsilon})(N+N_{CP})m} & \bar{Z}_{-k_2+i_1} &= \sum_{m=0}^{L-1} Z[-k_2+i_1, m] e^{-j(2CFO+2\pi\hat{\varepsilon})(N+N_{CP})m} \\ \bar{Z}_{-k_1+i_2} &= \sum_{m=0}^{L-1} Z[-k_1+i_2, m] e^{-j(2CFO+2\pi\hat{\varepsilon})(N+N_{CP})m} & \bar{Z}_{-k_2+i_2} &= \sum_{m=0}^{L-1} Z[-k_2+i_2, m] e^{-j(2CFO+2\pi\hat{\varepsilon})(N+N_{CP})m}\end{aligned}$$

等式 7 用於估計 I/Q 失衡相位估計的值，其使用等式 8 計算。

### 等式 7

$$B_n = \begin{cases} CFO \neq 0 & \frac{1}{2} \left( \frac{\sin(2 \cdot CFO \cdot N)}{CFO + \frac{\pi \cdot i_n}{N}} \right) + \frac{j}{2} \left( \frac{\cos(2 \cdot CFO \cdot N) - 1}{CFO + \frac{\pi \cdot i_n}{N}} \right) \\ CFO = 0, i_n = 0 & N \\ CFO = 0, i_n \neq 0 & 0 \end{cases}$$

$$\begin{aligned}C_1 &= \frac{e^{j2CFO(\Delta n)}}{N} \frac{(|B_1|^2 + |B_2|^2) \cdot \bar{Z}_{k_1}}{B_1(\bar{Z}_{-k_1+i_1}) + B_2(\bar{Z}_{-k_1+i_2})} \\ C_2 &= \frac{e^{j2CFO(\Delta n)}}{N} \frac{(|B_1|^2 + |B_2|^2) \cdot \bar{Z}_{k_2}}{B_1(\bar{Z}_{-k_2+i_1}) + B_2(\bar{Z}_{-k_2+i_2})} \\ C &= \frac{1}{2}(C_1 + C_2)\end{aligned}$$

### 等式 8

$$\overline{ge^{-j\theta}} = \frac{C-1}{C+1}$$

然後依據等式 9 計算 I/Q 失衡補償參數  $\xi$ 、 $\rho$  和 Scale\_Q(見例如等式 1)。

等式 9

$$\left\{ \begin{array}{ll} \text{real}\{ge^{-j\theta}\} \geq 1 & \text{ScaleQ} = 1 \\ \text{otherwise} & \text{ScaleQ} = 0 \end{array} \right. \quad \left\{ \begin{array}{l} \hat{\xi} = \frac{1}{\text{real}\{ge^{-j\theta}\}}, \hat{\rho} = -\frac{\text{imag}\{ge^{-j\theta}\}}{\text{real}\{ge^{-j\theta}\}} \\ \hat{\xi} = \text{real}\{ge^{-j\theta}\}, \hat{\rho} = -\text{imag}\{ge^{-j\theta}\} \end{array} \right.$$

等式 9 避免接收器處的飽和，因為  $\xi$  總是小於或等於一整數，因此削弱較強的 I/Q 信號，而不是放大較弱的 I/Q 信號。在一些實施例中，上述的計算可在幾個探測 2 傳輸上以疊代方式執行。等式 10 示出新的相位估計如何用於更新前一估計。

等式 10

$$\begin{aligned} \left(ge^{-j\theta}\right)_i^{\text{ACC}} &= (1 - \mu_i) \left(ge^{-j\theta}\right)_{i-1}^{\text{ACC}} + \mu_i \left(ge^{-j\theta}\right)_{i-1}^{\text{ACC}} \cdot \left(ge^{-j\theta}\right)_i \\ \left(ge^{-j\theta}\right)_0^{\text{ACC}} &= 1 \end{aligned}$$

在等式 10 中， $\left(ge^{-j\theta}\right)_i$  是在第  $i$  探測 2 傳輸中所計算的相位估計。一些實施例可包括更新程式，其可使用具有  $\mu_i \in [0,1]$  環增益的第一級環。環增益可提供收斂速度和環帶寬\_“BW”控制的噪音濾波之間的折衷。在收斂的過程中，環 BW 動態地改變中可使用變速\_gear-shifting 方法。對於快速收斂，在第一 2/3 疊代過程中，可使用高環 BW。對於連續的探測 2 傳輸，可使用小環 BW。等式 11 所列的值可用於  $\mu_i$   $i$  定義為探測 2 猝發係數。

等式 11

$$\mu_i = \begin{cases} 1 & i = 1 \\ 0.75 & i = 2 \\ 0.5 & i = 3 \\ 0.25 & i = 4 \end{cases}$$

等式 12 所列的 I/Q 補償參數可在第  $i$  探測 2 接收的過程中使用。

等式 12

$$\left\{ \begin{array}{l} \text{real}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\} \geq 1 \\ \text{otherwise} \end{array} \right. \hat{\xi}_i = \frac{1}{\text{real}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\}}, \hat{\rho}_i = -\frac{\text{imag}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\}}{\text{real}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\}}$$

$$\hat{\xi}_i = \text{real}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\}, \hat{\rho}_i = -\text{imag}\left\{\left(\overline{ge^{-j\theta}}\right)_{i-1}^{\text{ACC}}\right\}$$

對於 I/Q 失衡，3 到 4 個疊代(相當於 3 到 4 個探測 2 傳輸)常常是足夠補償的。

附錄 C 所列的虛擬碼用於補償的定點實施。

附錄 D 所列的參數用於 I/Q 失衡補償的系統中的硬體－軟體介面。

當網路節點處理一個或多個網路協調器(“NC”)輸出的探測 1 猝發時，根據每一信號音和載波頻率偏移(相對於一相關的網路協調器)，網路節點可獲得信噪比(“SNR”)估計。該節點可使用 SNR 估計告知 NC，哪兩個窗口中一個用於將探測 2 發送到該節點。該節點可使用 CFO 估計計算 OFDM 符號的數量，及探測 2 傳輸過程中迴圈字首(“CP”)的長度，並發送到 NC。

附錄 E 所列的虛擬碼，用於計算頻率偏移引入、CP 以及 OFDM 符號數量的選擇。在一些實施例中，附錄 E 所列的探測 2、CP 以及 L 演算法可在發送 MoCATM 探測 2 報告之前和接收器 RF 發生器引入任意需要的、必要的或人為的載波偏移之後執行。

為了清楚的緣故，前面的描述，包括參數或參數值的特定例子，有時是針對一特定的協定，例如那些標識名稱為 MoCATM 和 /或乙太網協定。然而，不限於此，本發明可普遍到別的協定和 /或資料包協定。所使用的術語是針對一特定的協定，例如其標識為 MoCATM 或乙太網，以描述一特定特點或實施例，但不是特別地用於限制該特點或實施例的範圍於該協定；取代通常所使用的術語，並包括每一個在別的協定下所定義的平行和類似的術語。

可以理解的，本發明的軟體元件包括程式和資料，如果期望

的話，可以 ROM(唯讀記憶體)形式，包括 CD-ROM、EPROM 和 EEPROM，進行實施，或可存儲在任意合適的電腦可讀媒介，例如但不限於各種不同的光碟、各種不同的磁帶和 RAM。在此描述為軟體的元件可選擇地、全部或部分地實施在硬體中，如果期望的話，可使用傳統技術。

因此，已對用於補償 I/Q 失衡的系統和方法進行了描述。本發明是通過一些實施例進行描述的，本領域技術人員知悉，在不脫離本發明的精神和範圍的情況下，可以對這些特徵和實施例進行各種改變或等效替換。本發明不僅僅限於如下的權利要求。

探測 2 理論 I/Q 失衡可類比為倍增的增益因數，用於 I/Q 分量的其中之一，也可類比為相對的相位差。在探測 2 接收的過程中，MoCA 規定：在 RF 下變頻過程中，接收器必須引入頻率誤差，我們定義該偏移為  $\varphi$ 。經轉換的信號為：

$$\begin{aligned} z_i[n] &= s_i[n]\cos(2\pi\varphi n) - s_q[n]\sin(2\pi\varphi n) + w_i[n] \\ z_q[n] &= g s_i[n]\sin(2\pi\varphi n - \theta) + g s_q[n]\cos(2\pi\varphi n - \theta) + w_q[n] \end{aligned}$$

一些代數式可將上述表達為：

$$\begin{aligned} z'[n] &= \left\{ \text{real} \left( K_1 s[n] e^{j2\pi\varphi n} + K_2 s^*[n] e^{-j2\pi\varphi n} + w[n] \right) \right\} \\ &+ j\xi \left\{ \text{imag} \left( K_1 s[n] e^{j2\pi\varphi n} + K_2 s^*[n] e^{-j2\pi\varphi n} + w[n] \right) \right\} \\ &+ j\rho \left\{ \text{real} \left( K_1 s[n] e^{j2\pi\varphi n} + K_2 s^*[n] e^{-j2\pi\varphi n} + w[n] \right) \right\} \end{aligned}$$

在接收器，執行 I/Q 補償，I/Q 補償後的信號為：

假設以頻率視窗  $k$ ，發送單一頻率，經幾個代數過程後的補償信號為：

$$\begin{aligned} z'[n] &= \left\{ |h| \cos \left( 2\pi n \left( \frac{k}{N} + \varphi \right) + \angle h \right) \right\} \\ &+ j\xi \left\{ g |h| \sin \left( 2\pi n \left( \frac{k}{N} + \varphi \right) + \angle h - \theta \right) \right\} \\ &+ j\rho \left\{ |h| \cos \left( 2\pi n \left( \frac{k}{N} + \varphi \right) + \angle h \right) \right\} + [w_r[n] + j(\rho \cdot w_r[n] + \xi \cdot w_i[n])] \end{aligned}$$

信號經過頻率補償，然後經過 FFT 操作轉換到頻域。經過幾個代數過程後在視窗  $k$  和  $-k$  的頻域信號為：

$$\hat{Z}[k] = |h|e^{j\omega k} N \left[ \frac{1}{2}(1 + \xi g e^{-j\theta}) + \frac{\rho}{2} e^{j\frac{\theta}{2}} \right] + |h|e^{-j\omega k} e^{-j(2\pi(N-1)(\frac{k}{N} + \varphi))} \frac{\sin(2\pi(k + N\varphi))}{\sin(2\pi(\frac{k}{N} + \varphi))} \left[ \frac{1}{2}(1 - \xi g e^{j\theta}) + \frac{\rho}{2} e^{j\frac{\theta}{2}} \right]$$

$$\hat{Z}[-k] = |h|e^{-j\omega k} e^{-j(2\pi(N-1))} \frac{\sin(2\pi\varphi N)}{\sin(2\pi\varphi)} \left\{ \frac{1}{2}(1 - \xi g e^{j\theta}) + \frac{\rho}{2} e^{j\frac{\theta}{2}} \right\}$$

在沒有 I/Q 失衡的系統中，在負視窗的能量為零。由於 I/Q 失衡，在負窗口的能量為：

$$|\hat{Z}[-k]|^2 = |h|^2 \left( \frac{\sin(2\pi\varphi N)}{\sin(2\pi\varphi)} \right)^2 \left\{ \frac{1}{2}(1 - \xi g e^{j\theta}) + \frac{\rho}{2} e^{j\frac{\theta}{2}} \right\} \left\{ \frac{1}{2}(1 - \xi g e^{-j\theta}) + \frac{\rho}{2} e^{-j\frac{\theta}{2}} \right\}$$

$$= \frac{1}{4} |h|^2 \left( \frac{\sin(2\pi\varphi N)}{\sin(2\pi\varphi)} \right)^2 \{ 1 + \xi^2 g^2 + \rho^2 - 2\xi g \cos\theta - 2\xi g \rho \sin(\theta) \}$$

因此，我們的目標是使用  $\rho$ ， $\xi$ ，最小化窗口  $-k$  的能量。使用拉格朗日乘數法進行最小化，給出如下等式

$$\frac{\partial |\hat{Z}[-k]|^2}{\partial \xi} = \frac{\partial (1 + \xi^2 g^2 + \rho^2 - 2\xi g \cos\theta - 2\xi g \rho \sin(\theta))}{\partial \xi} = 0$$

$$\frac{\partial |\hat{Z}[-k]|^2}{\partial \rho} = \frac{\partial (1 + \xi^2 g^2 + \rho^2 - 2\xi g \cos\theta - 2\xi g \rho \sin(\theta))}{\partial \rho} = 0$$

$$2\xi g^2 - 2g \cos\theta - 2\rho g \sin(\theta) = 0$$

$$2\rho - 2\xi g \sin(\theta) = 0$$

解出上面的量

$$\begin{bmatrix} g^2 & -g \sin(\theta) \\ -g \sin(\theta) & 1 \end{bmatrix} \begin{bmatrix} \xi \\ \rho \end{bmatrix} = \begin{bmatrix} g \cos\theta \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \xi \\ \rho \end{bmatrix} = \frac{1}{(g \cos\theta)^2} \begin{bmatrix} g \cos\theta \\ g^2 \sin\theta \cos\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{g \cos\theta} \\ \tan\theta \end{bmatrix}$$

容易得出，這樣的選擇實際上使得窗口  $-k$  的能量等於 0，因此完全消除了 I/Q 失衡的影響。現在，我們的目的是從探測 2 傳輸估計出 I/Q 失衡參數。

### I/Q 參數估計

因為 I/Q 失衡破壞了輸入的信號，所以其導致了受到破壞的載波頻率估計，及受到破壞的通道估計。在 I/Q 失衡下的通道估計為：

$$\hat{h}_k = \frac{A_k}{N|A_k|^2} \hat{Z}[k] = |h|e^{j\alpha} K_1 + e_k$$

在視窗  $k$  和  $-k$ ，FFT 的輸出不具有 I/Q 失衡補償，但是在頻率補償後，假設  $\varepsilon$  的頻率估計誤差為：

$$Z[k] = |h|e^{j\alpha} K_1 e^{-j2\pi(N-1)\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right)} \frac{\sin(2\pi N\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right))}{\sin(2\pi\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right))} + W_k$$

$$Z[-k] = |h|e^{-j\alpha} K_2 e^{-j2\pi(N-1)\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right)} \frac{\sin(2\pi N\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right))}{\sin(2\pi\left(\frac{k}{N} + \varphi - \frac{1}{2}\varepsilon\right))} + W_{-k}$$

### 載波頻率偏移大於 50KHz 的影響

在載波頻率誤差出現的情況下，由 I/Q 失衡所導致的鏡像分量(image component)正好在所發送的信號音的鏡像數位頻率( $-k/N$ )出現。在載波頻率誤差(在探測 II 過程中是強制的)下，I/Q 鏡像在 $(-k/N - 2\varphi)$ 數位頻率出現，其中  $\varphi$  是標準化的載波頻率誤差， $\varphi = \text{載波頻率誤差}/\text{符號率} = \Delta f_c/f_s$ 。在  $1.5e9\text{Hz} = 300\text{kHz}$  下，載波頻率誤差的範圍可在  $\pm 200\text{ppm}$  之間，而 OFDM 信號音間隔為  $50e6/256 = 195.3\text{kHz}$ 。因此，在頻域中，鏡像分量可落在  $[-k-3, k+3]$  的時間間隔之間。對於窗口  $-k+i$ ，FFT 輸出為：

$$z[-k+i] = |h|e^{-\alpha} K_2 e^{-j2\pi(\varphi - \frac{1}{2}\varepsilon)(N-i)} \left( \frac{(-1)^i \sin(2\pi N[\varphi - \frac{1}{2}\varepsilon])}{\sin(2\pi[\varphi - \frac{1}{2}\varepsilon + \frac{1}{2N}])} \right) + |h|e^{-\alpha} K_1 e^{-j\pi(\frac{M+i}{N} + \varepsilon)(N-i)} \left( \frac{\sin(2\pi N[\frac{M+i}{N} + \varepsilon])}{\sin(\pi[\frac{M+i}{N} + \varepsilon])} \right) + W_{-k+i}$$

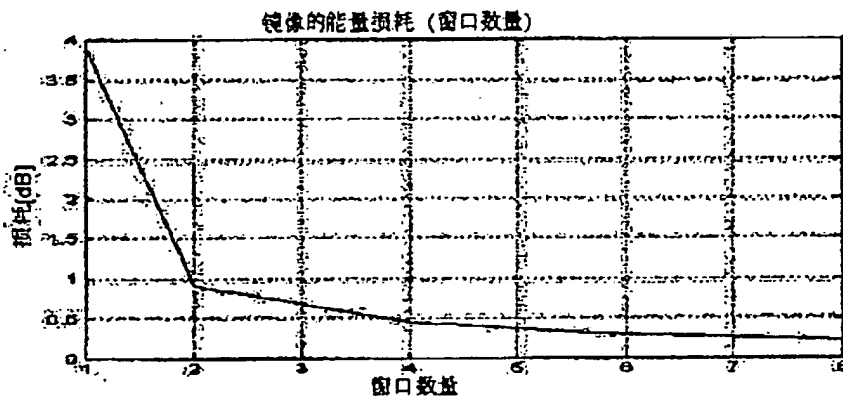
並且由於補償頻率誤差  $\varphi$  導致在鏡像位置有  $2\varphi$  的偏移，我們需要從時間間隔  $[-k-3, \dots, k+3]$  收集鏡像能量。實際地，因為我們知道頻率誤差  $\varphi$  (等於  $\varepsilon$ )，所以我們知道鏡像將以以下的數位頻率出現：

$$-\left(\frac{k+M}{N} + r\right), 2\varphi = \frac{M}{N} + r \text{ where } M = \lfloor 2\varphi N \rfloor \text{ and } |r| < \frac{1}{N}.$$

以 [dB] 表示的鏡像能量的損耗是用於收集能量的視窗的數量的函數，其為：

$$Loss = 10 \log_{10} N^2 - 10 \log_{10} \left( \sum_{i=-M-L}^{-M+L} \left( \frac{\sin(\pi[M+Nr])}{\sin\left(\pi\left[\frac{M+i}{N} + r\right]\right)} \right)^2 \right)$$

最壞的損耗情況是當鏡像落在視窗 ( $r = 1/(2N)$ ) 之間。使用最接近鏡像的窗口將導致 3.9223 [dB] 的最壞損耗；使用兩個窗口導致 0.9120 [dB] 的損耗。下圖總結了作為所使用的視窗的數量的函數的損耗。



我們使用 2 個窗口，似乎是負責度和執行之間合理的折衷。  
探測 2 的 FFT 處理 (單 OFDM 符號)

為了簡單起見，考慮單一 OFDM 符號，稍後將給出擴展到多 OFDM 符號。我們已經示出在點  $k$  和  $-k$  的 FFT 輸出為：

$$\left. \begin{aligned} Z[k] &= \overbrace{A_1 h K_1}^{\text{信号项}} + \underbrace{B_1 h^* K_2}_{\text{来自窗口-k的ICI}} + W_1 \\ Z[-k] &= \underbrace{A_2 h K_1}_{\text{来自窗口k的ICI}} + \overbrace{B_2 h^* K_2}^{\text{信号项}} + W_2 \end{aligned} \right|$$

容易看出，每一運算式中包括期望信號項和來自鏡像頻率的 ICI 項。現在，我們將示出 ICI 項遠小於信號項，並可忽略。

在視窗  $k$  所引發的 ICI，是由於 I/Q 失衡所導致的鏡像信號以  $-\frac{k}{N} - \varphi'$  數位頻率所產生  $\varphi' = (\varphi - \frac{1}{2}\varepsilon)$ ，且該數位頻率不位於 FFT 刪格上。該頻率離 FFT 刪格越遠，ICI 就越大。因為  $k$  限制在時間間隔  $\{[146,186],[217,249]\}$ ，因此，所生成的鏡像與期望的信號差別很大，並且以頻率生成的 ICI 噪音就非常小。為了瞭解這個，要考慮在視窗  $k$  的信號和 ICI 項之間的比率。我們定義該比率為期望和 ICI 項之間的 SNR，為：

關於  $\{g, \theta, k, \varphi'\}$ ，通過最小化上面的運算式，將得到最壞的 SNR。容易得知，最小化如上運算式是可分離的，因此可通過如下實現最小化：

- 在約束條件  $g \in [0.5, 2]$  (最多 3[dB] 振幅失衡)，以及  $\theta \in [-10^\circ, 10^\circ]$  下，最小化關於  $g$  和  $\theta$  的第一項

- 在約束條件  $\varphi' \in [-200e - 6 * 1.5e9 / 50e6 : -200e - 6 * 1.5e9 / 50e6]$ ，以及  $k \in [146 : 186, 217 : 249]$  下，最大化第二項

可使用 Matlab 仿真數位化地執行以上的最小化操作。下圖將第一項作為  $g$  和  $\theta$  的函數進行描述

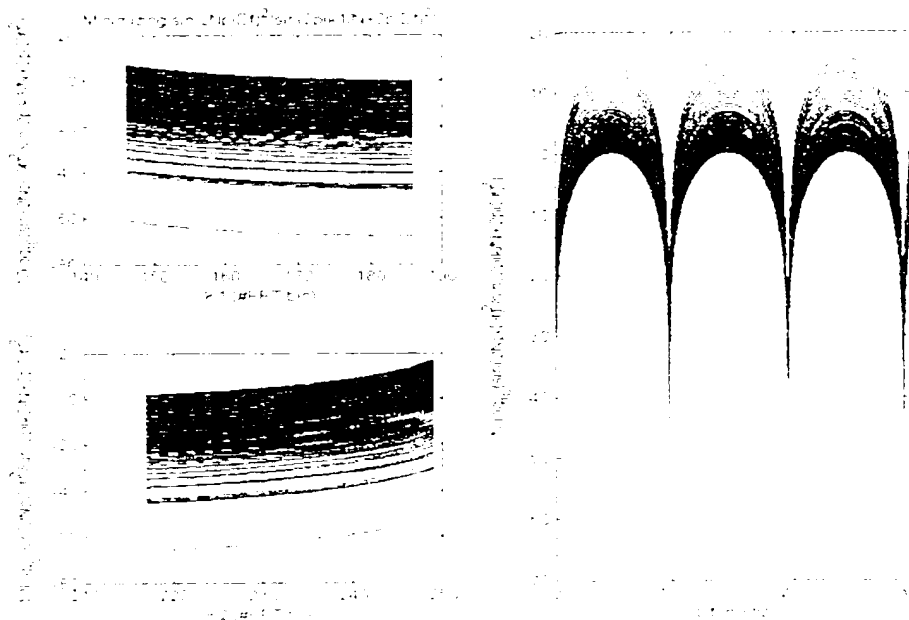


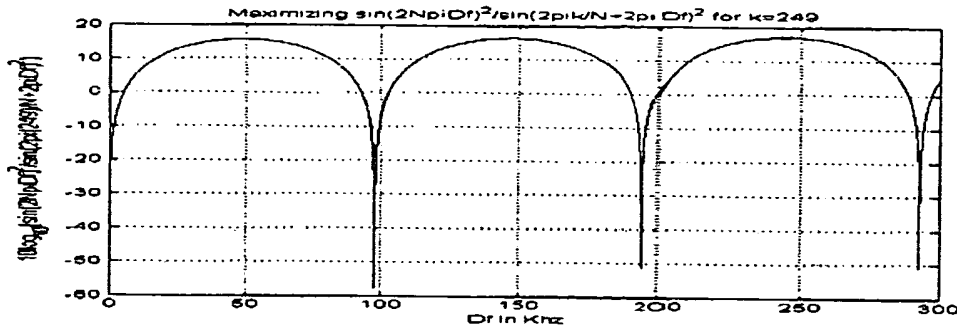
容易看出(也可分析得到), 最小值在多數時間間隔的邊界, 即  $g \in [0.5, 2]$  且  $Teta \in [10, 10]$ , 因此

$$57.4216 = \min_{g \in [0.5, 2], \theta \in [10, 10]} 10 \log_{10} \frac{N^2 K_1(g, \theta)}{K_2(g, \theta)}$$

下圖將第二項作為  $\theta$  和  $k$  的函數進行描述

從上圖, 容易看出, 在  $k = 249$  時, 第二項取得最大值, 因此對所得的第二項進行了描述





當  $Df = \pm 245 \text{ KHz}$  時取得最大值，因此

$$17.04 = \max_{g, \theta} \left\{ 10 \log_{10} \left( \left[ \frac{\sin(2\pi\phi N)}{\sin\left(\frac{2\pi\phi}{N} + 2\pi\phi\right)} \right]^2 \right) \right\} \quad \text{st} \quad \begin{array}{l} \phi \in [-\Delta f, \Delta f] \\ k \in [146, 186] \cup [217, 249] \end{array}$$

所以，由 ICI 項所導致的 SNR 最壞的情況為 40.3816[dB]

$$40.3816[\text{dB}] = \min(\text{SNR}_k) = \min_{\phi, k, g, \theta} \left( 10 \log_{10} \left( \frac{N^2 |K_1(g, \theta)|^2}{|K_2(g, \theta)|^2} \right) - 10 \log_{10} \left( \left[ \frac{\sin(2\pi\phi N)}{\sin\left(\frac{2\pi\phi}{N} + 2\pi\phi\right)} \right]^2 \right) \right)$$

因此，ICI 項最壞為 40[dB]，小於信號項，所以可以忽略。對負窗口可以進行類似的分析。在忽略 ICI 項後，在窗口  $k$  和  $-k+i$ ，FFT 的輸出為：

$$\begin{aligned} Z[k] &= hAK_1 + W_k \\ Z[-k+i] &= h^* B_i K_2 + W_{-k+i} \\ A &= e^{j\pi\epsilon(N-1)} \frac{\sin(\pi\epsilon N)}{\sin(\pi\epsilon)} \\ B_i &= e^{-j2\pi(\phi - \frac{1}{2}\epsilon + \frac{1}{2N})(N-1)} \left( \frac{(-1)^i \sin(2\pi N[\phi - \frac{1}{2}\epsilon])}{\sin(2\pi[\phi - \frac{1}{2}\epsilon + \frac{1}{2N}])} \right) \end{aligned}$$

因為我們不能估計通道回應  $h$ ，我們不能解決  $ge^{-j\theta}$  的線性 LS 問題，所以我們先從兩個負窗口  $-k+i_1$  和  $-k+i_2$ ，解決  $hK_2^*$  估計的 LS 問題

$$\begin{bmatrix} \hat{Z}^*[-k+i_1] \\ \hat{Z}^*[-k+i_2] \end{bmatrix} = \begin{bmatrix} B_1^* \\ B_2^* \end{bmatrix} (hK_2^*) + \begin{bmatrix} W_{k+i_1}^* \\ W_{k+i_2}^* \end{bmatrix}$$

$$(hK_2^*)_{LS} = \frac{B_1 \hat{Z}^*[-k+i_1] + B_2 \hat{Z}^*[-k+i_2]}{|B_1|^2 + |B_2|^2}$$

在不知道通道  $h$  的情況下，我  $\frac{K_1}{K_2}$  們能估計

$$C \equiv \frac{\overline{K_1}}{K_2^*} = \frac{(hK_1)}{(hK_2^*)_{LS}} = \frac{z[k]_A}{\frac{B_1 \hat{Z}^*[-k+i_1] + B_2 \hat{Z}^*[-k+i_2]}{|B_1|^2 + |B_2|^2}} = \frac{(|B_1|^2 + |B_2|^2) \cdot Z[k]}{A \cdot (B_1 \hat{Z}^*[-k+i_1] + B_2 \hat{Z}^*[-k+i_2])}$$

因為探測 2 是由兩個信號音組成，其中一個在  $k_1$ ，另一個在  $k_2$ ，因此根據這兩個信號音，平均該結果，得

$$\bar{C} \equiv \frac{1}{2} \frac{(|B_1|^2 + |B_2|^2) \cdot Z[k_1]}{A \cdot (B_1 \hat{Z}^*[-k_1+i_1] + B_2 \hat{Z}^*[-k_1+i_2])} + \frac{1}{2} \frac{(|B_1|^2 + |B_2|^2) \cdot Z[k_2]}{A \cdot (B_1 \hat{Z}^*[-k_2+i_1] + B_2 \hat{Z}^*[-k_2+i_2])}$$

容易看出

$$ge^{-j\theta} = \frac{K_1/K_2^* - 1}{K_1/K_2^* + 1}$$

因此其估計為

$$\overline{ge^{-j\theta}} = \frac{\bar{C} - 1}{\bar{C} + 1}$$

接著可以計算出 I/Q 估計

$$\hat{\xi} = \frac{1}{\text{real}\{ge^{-j\theta}\}}$$

$$\hat{\rho} = -\frac{\text{imag}\{ge^{-j\theta}\}}{\text{real}\{ge^{-j\theta}\}}$$

### 探測 2 的 FFT 處理 (多 OFDM 符號)

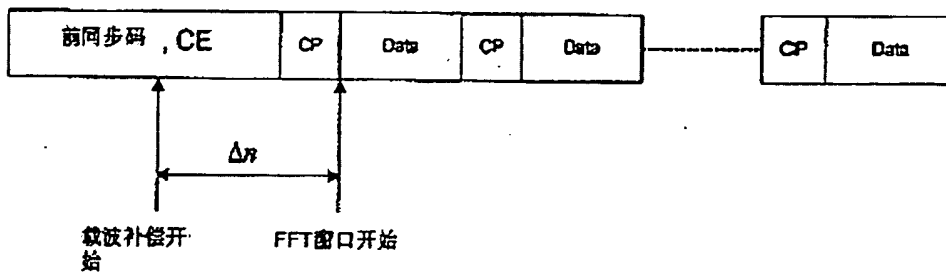
當我們注意多 OFDM 符號時，我們需要考慮由殘餘頻率誤差  $\varepsilon$  所導致的相位誤差。容易示出第  $m$  個 OFDM 符號相對於第一個

的相位為

$$Z[k,m] = \begin{cases} k \geq 0 & e^{j2\pi(N+N_{CP})m} Z[k,0] \\ k < 0 & e^{j2\pi(2p-\epsilon)(N+N_{CP})m} Z[k,0] \end{cases}$$

注意到上式沒有考慮採樣頻率誤差，假設其影響很小，所以在整個分析過程中忽略。

從載波頻率補償開始到第一 FFT 視窗開始，所累加的相位應該考慮到。因為我們的演算法是計算  $Z_k$  和  $\text{conj}(Z_{-k})$  之間的比率，所以任意的常數相位項不僅不能刪除，而且要將其加倍。



因此，對於第  $m$  OFDM 符號，在視窗  $+k$ 、 $-k+i$ ，FFT 輸出設為

$$Z[k] = D \underbrace{e^{-j2\pi k(p-\epsilon)}}_{\#1} \underbrace{e^{j2\pi(N+N_{CP})m}}_{\#2} h A K_1 + W_k$$

$$Z[-k+i] = D' \underbrace{e^{-j2\pi k(p-\epsilon)}}_{\#1} \underbrace{e^{j2\pi(2p-\epsilon)(N+N_{CP})m}}_{\#2} h' B_i K_1 + W_{-k+i}$$

#1- 由于补偿开始与FFT之间的 $m$ 样本不能取消  
 #2- 由于补偿开始与FFT开始之间的 $m$ 样本不能取消  
 #3- 相位由于残余载波频率误差

殘餘載波頻率估計

為了使用從所有  $L$  OFDM 符號中得出的資訊，我們需要補償殘餘頻率偏移  $\epsilon$ ，接著為每一個視窗，計算補償信號的平均數，以降低 AWGN 方差。因為  $\epsilon$  可以足夠大，以便在  $L$  OFDM 符號期間，可發生多次相位纏繞(phase wrapping)。我們提出的如下估計器，其不受相位纏繞的影響(只要在兩個連續的樣本之間發生一次纏

繞，就運用該估計器)

### 殘餘頻率估計

可通過計算得到殘餘頻率誤差估計

$$\hat{\varepsilon} = \frac{\text{angle}\left(\sum_{m=0}^{L-2} Z[k_i, m]Z^*[k_i, m+1]\right)}{2\pi(N + N_{CP})}$$

其中

$$k_i = \begin{cases} k_1 & SNR_{k_1} > SNR_{k_2} \\ k_2 & SNR_{k_2} > SNR_{k_1} \end{cases}$$

殘餘頻率誤差和時間平均設為

殘餘頻率補償和時間平均

$$\begin{aligned} \bar{Z}_{k_1} &= \sum_{m=0}^{L-1} Z[k_1, m]e^{-j2\pi\varepsilon(N+N_{CP})m} & \bar{Z}_{k_2} &= \sum_{m=0}^{L-1} Z[k_2, m]e^{-j2\pi\varepsilon(N+N_{CP})m} \\ \bar{Z}_{-k_1+i_1} &= \sum_{m=0}^{L-1} Z[-k_1+i_1, m]e^{-j2\pi(2\hat{\omega}+\varepsilon)(N+N_{CP})m} & \bar{Z}_{-k_2+i_1} &= \sum_{m=0}^{L-1} Z[-k_2+i_1, m]e^{-j2\pi(2\hat{\omega}+\varepsilon)(N+N_{CP})m} \\ \bar{Z}_{-k_1+i_2} &= \sum_{m=0}^{L-1} Z[-k_1+i_2, m]e^{-j2\pi(2\hat{\omega}+\varepsilon)(N+N_{CP})m} & \bar{Z}_{-k_2+i_2} &= \sum_{m=0}^{L-1} Z[-k_2+i_2, m]e^{-j2\pi(2\hat{\omega}+\varepsilon)(N+N_{CP})m} \end{aligned}$$

使用如上相同的估計器設備，接著可估計向量  $ge^{-j\theta}$ ，即

$$C_i = e^{j4\pi(\varphi-\varepsilon)(\Delta n)} \frac{(|B_1|^2 + |B_2|^2) \cdot \bar{Z}_{k_i}}{B_1(\bar{Z}_{-k_1+i_1})^* + B_2(\bar{Z}_{-k_1+i_2})^*}$$

其中，相位項  $e^{j4\pi(\varphi-\varepsilon)(\Delta n)}$  用於補償從頻率補償開始時間直到第一 FFT 視窗開始所累加的初始相位誤差。

簡化係數  $B_i$  和  $A$  對於實際執行，我們需要簡化  $B_i$  和  $A$  的運算式；通過引入一些近似值，可以實現簡化。讓我們看

$$A = e^{j\pi\varepsilon(N-1)} \frac{\sin(\pi\varepsilon N)}{\sin(\pi\varepsilon)}$$

殘餘頻率誤差通常小於 10kHz(7ppm)，對這樣的一個誤差

$$e^{j\pi\epsilon(N-1)} = 0.9872 + 0.1595i \cong 1$$

$$\frac{\sin(\pi\epsilon N)}{\sin(\pi\epsilon)} = 254.8975 \cong 256 = N$$

因此我們可以作如下的近似

$$A = e^{j\pi\epsilon(N-1)} \frac{\sin(\pi\epsilon N)}{\sin(\pi\epsilon)} \cong N$$

對於  $B_i$

$$B_i = e^{-j2\pi(\varphi - \frac{1}{2}\epsilon + \frac{i}{2N})(N-1)} \left( \frac{(-1)^i \sin(2\pi N[\varphi - \frac{1}{2}\epsilon])}{\sin(2\pi[\varphi - \frac{1}{2}\epsilon + \frac{i}{2N}])} \right)$$

$$\cong e^{-j2\pi(\varphi - \frac{1}{2}\epsilon)N} e^{-j\pi} \left( \frac{(-1)^i \sin(2\pi N[\varphi - \frac{1}{2}\epsilon])}{\sin(2\pi[\varphi - \frac{1}{2}\epsilon + \frac{i}{2N}])} \right)$$

$$\cong e^{-j2\pi N(\varphi - \epsilon)} (-1)^{2i} \left( \frac{\sin(2\pi N[\varphi - \epsilon])}{\sin(2\pi[\varphi - \epsilon + \frac{i}{2N}])} \right)$$

$$= e^{-j2\pi N(\varphi - \epsilon)} \left( \frac{\sin(2\pi N[\varphi - \epsilon])}{\sin(2\pi[\varphi - \epsilon + \frac{i}{2N}])} \right)$$

其中  $\varphi - \epsilon = \frac{CFO}{2\pi}$

$$B_i \cong e^{-j2\pi N(\varphi - \epsilon)} \left( \frac{\sin(2\pi N[\varphi - \epsilon])}{\sin(2\pi[\varphi - \epsilon + \frac{i}{2N}])} \right)$$

$$= \frac{\sin(2 \cdot N \cdot CFO) + j[\cos(2 \cdot N \cdot CFO) - 1]}{2 \sin(2\pi[\varphi - \epsilon + \frac{i}{2N}])}$$

因為伴隨殘餘頻率誤差的頻率偏移小於  $(200 + 7)\text{ppm}$ ，並且在  $-3i3$ ，分母中的  $\sin$  的引數是較小的。

對於這樣小的角度，簡單的線性近似只有很小的誤差

因此接著

$$\sin(0.0121) \cong 0.0121$$

↓

$$\sin(2\pi[\varphi - \epsilon + \frac{i}{2N}]) \cong 2\pi[\varphi - \epsilon + \frac{i}{2N}]$$

因此接著

應此簡化的係數為

$$A = N$$

$$B = \frac{1}{2} \frac{\sin(2 \cdot CFO \cdot N)}{CFO - \frac{\pi}{2N}} + \frac{j}{2} \frac{\cos(2 \cdot CFO \cdot N) - 1}{CFO - \frac{\pi}{2N}}$$

$$B = \frac{1}{2} \frac{\sin(2 \cdot CFO \cdot N)}{CFO + \frac{\pi}{2N}} + \frac{j}{2} \frac{\cos(2 \cdot CFO \cdot N) - 1}{CFO + \frac{\pi}{2N}}$$

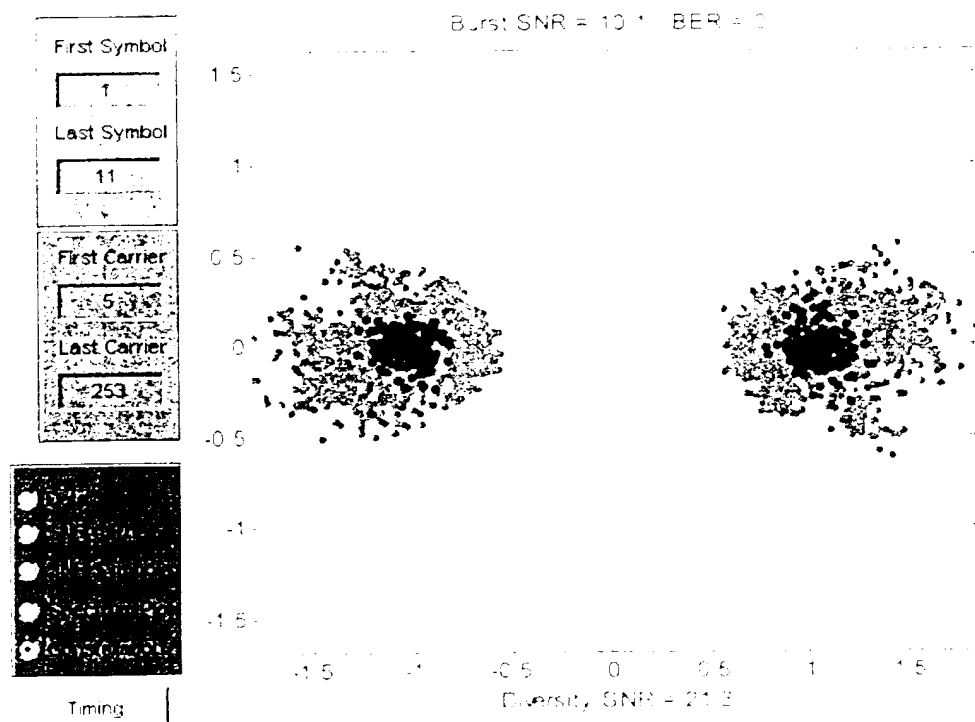
附錄 B

示範性度量

具有 200ppm 載波和採樣速率偏移的理想通道，無 AWGN

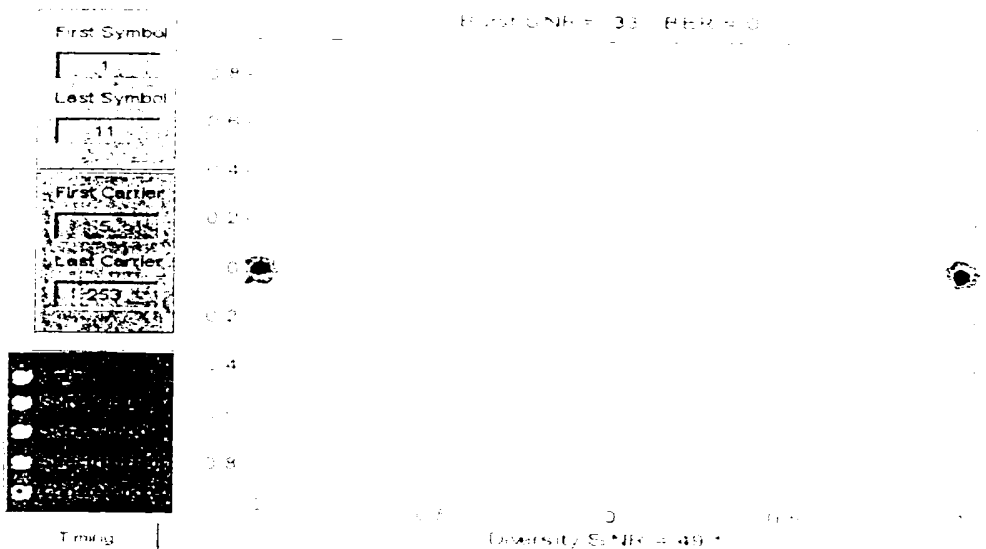
圖總結了 3[dB] 振幅失衡、10° 相位失衡、200ppm 頻率偏移、理想通道和沒有 AWGN 的仿真結果。在調用 RX\_I/Q 抽樣程式之前，接收器 SNR 大約為 10.1[dB]，從下圖 B-1 可以看出。

圖 B-1：在取碼之前，限幅器(slicer)輸入理想通道、無 AWGN



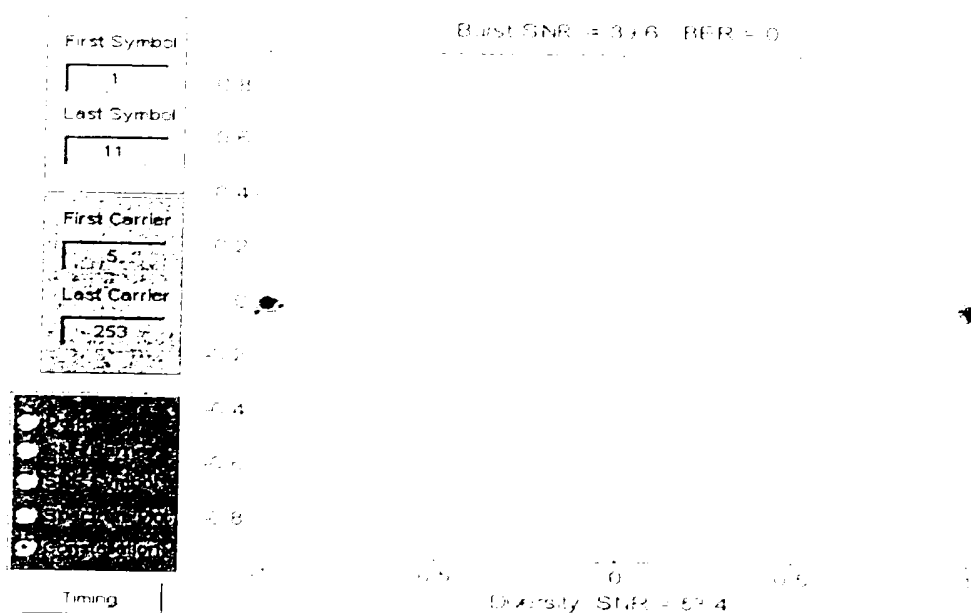
在處理第一採碼 2 後，SNR 大約為 3.3[dB]。在如圖 B-2 中，可在第一次疊加之前和之後，比較圖像信號(image signal)的大小。

圖 B-2：在第一次疊代之後，頻率輸入理想通道無 AWGN



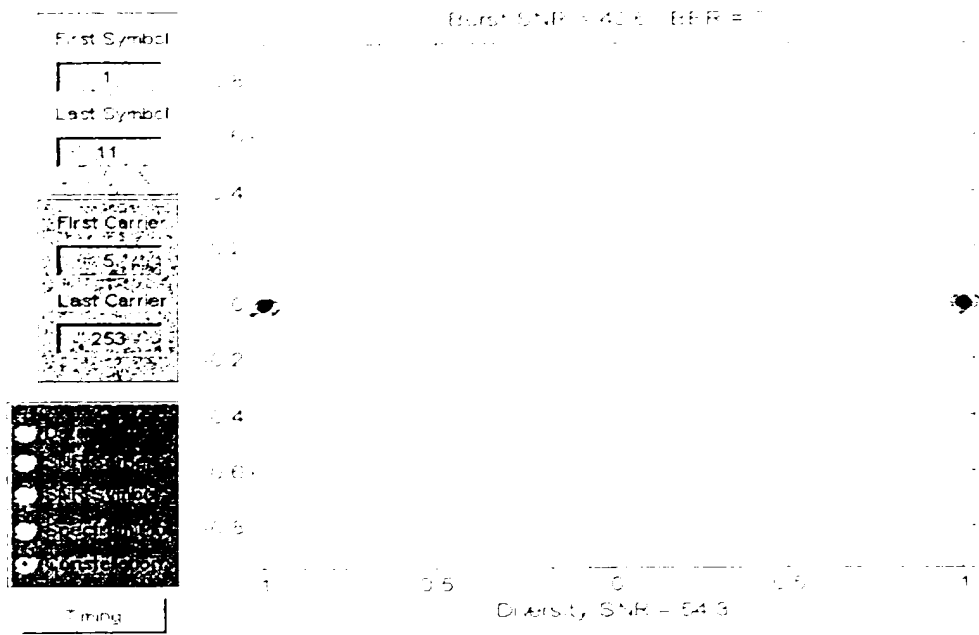
在處理第一探測 II 之後，SNR 大約為 39.6[dB]。在如下圖 B-3 中，可在第二次疊代之前和之後，比較鏡像信號的大小。在第二次疊代後，該鏡像信號不再可見。

圖 B-3：在兩次疊代之後，頻率圖理想通道、沒有 AWGN



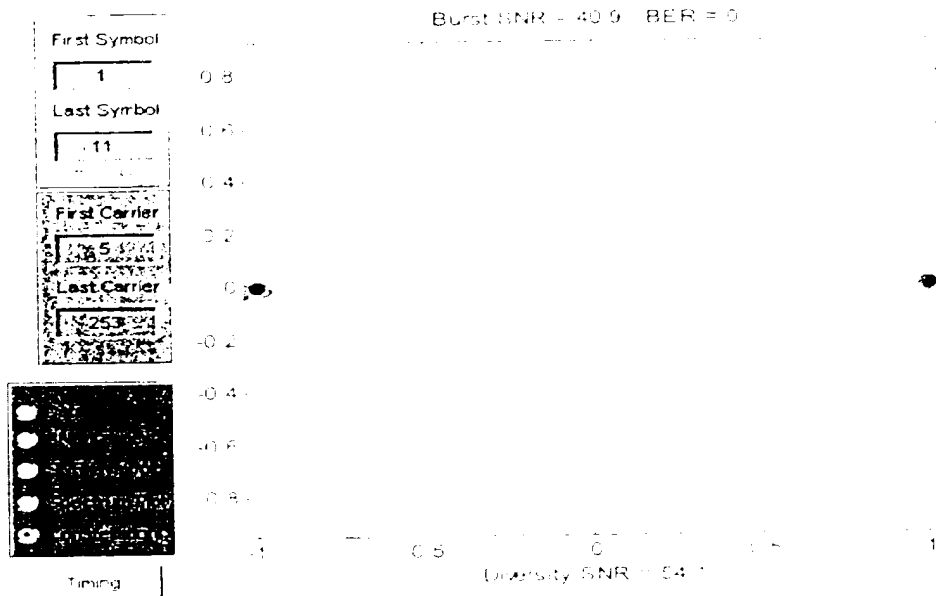
在處理第二探測 II 後，SNR 大約為 40.6[dB]

圖 B-4：在三次疊代之後，限幅器輸入理想通道、沒有 AWGN



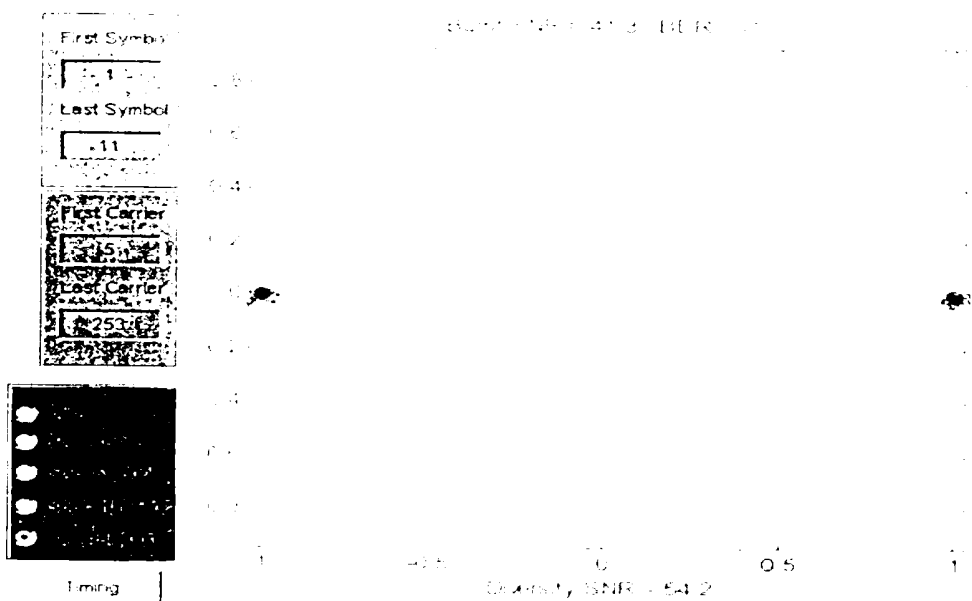
如圖 B-5 描述了限幅器 SNR，在處理了第四探測 II 之後，傳輸 SNR 則大約為 40.9[dB]

圖 B-5：在四次疊代之後，限幅器輸入理想通道、沒有 AWGN



如圖 B-6 描述了 I/Q 平衡的限幅 SNR，其 SNR 大約為 41.3[dB]。因此，將該 SNR 與圖 B-5 探測 II 傳輸後得到的 SNR 相比，我們推測出，相對於 41.3[dB] 的本底雜音(noise floor)，幾何 I/Q 平衡的表現雜音則大約為 0.4[dB]

圖 B-6：限幅器輸入理想通道，沒有 AWGN，沒有 I/Q 失衡

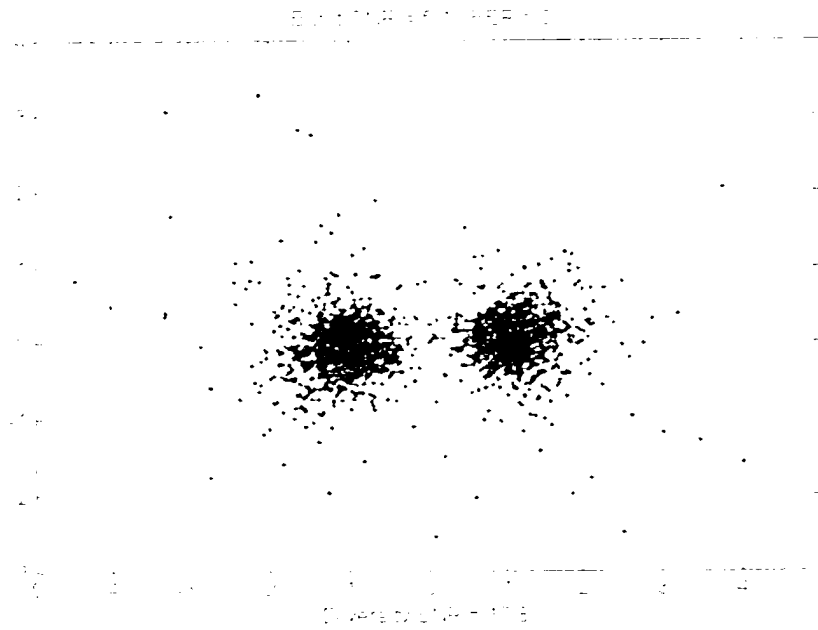


表格總結了，在四次疊代中每一次結束之後，I/Q 失衡參數是

疊代次數	增益	Theta [°]	限幅器 SNR [dB]
<i>True Value</i>	<i>0.70795</i>	<i>10</i>	<i>41.3</i>
#0	1	0	10.1
#1	0.71345	11.4402	33
#2	0.71103	10.3227	39.6
#3	0.70989	10.1742	40.6
#4	0.70944	10.1358	40.9

通道 MoCA10408，SNR AWGN 15[dB]。表格總結了 3[dB] 的增益失衡，10° 的相位失衡，200ppm 的頻率偏移。MoCA10408 頻道型 15[dB]AWGN 的仿真結果。在調用 RX I/Q 抽樣塊之前，接收器 SNR 大約為 5.1[dB]，參見圖 B-7 可以看見。

圖 B-7：在取消之前，限幅器輸入 MoCA10408 通道，15[dB]AWGN SNR



在處理第一探測 II 之後，SNR 大約為 9.5[dB]。在如下圖 B-8 中，可在第一疊代之前和之後，比較圖像信號的大小。

圖 B-8：在第一次疊代之後，頻率圖 MoCA10408 通道、15[dB]AWGN SNR

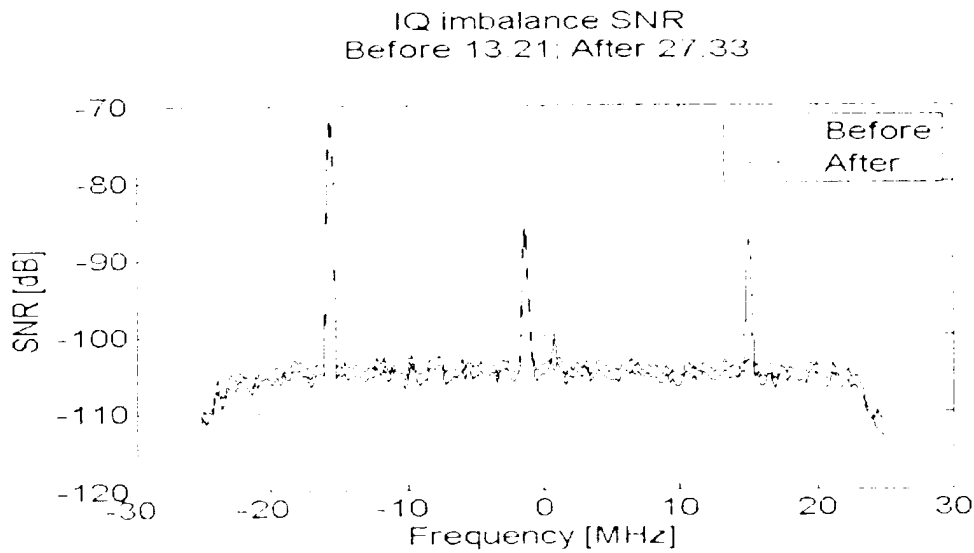


圖 B-9：在第一次疊代之後，限幅器輸入、MoCA10408 通道、15[dB]AWGN SNR

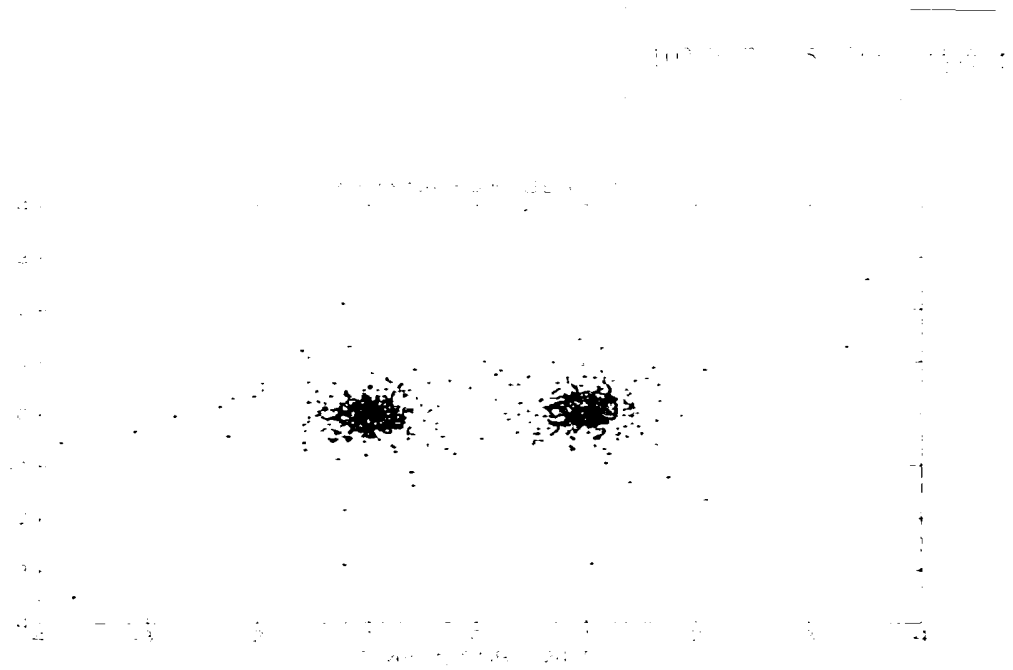


圖 B-10：在第二次疊代之後，SNR 收斂到 11[dB]

圖 B-10：在第二次疊代之後，輸入訊號，MoCA10408 頻道，15[dB]AWGN SNR

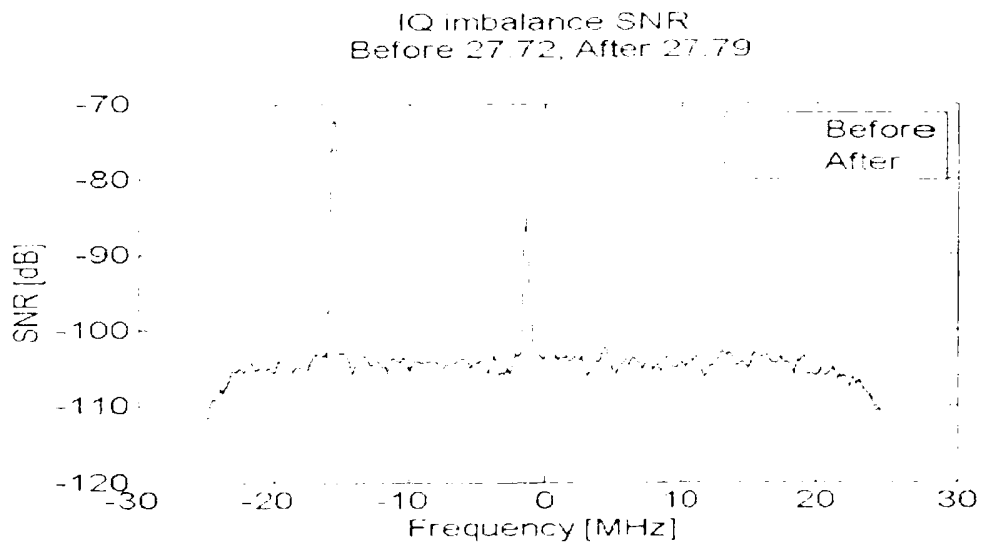
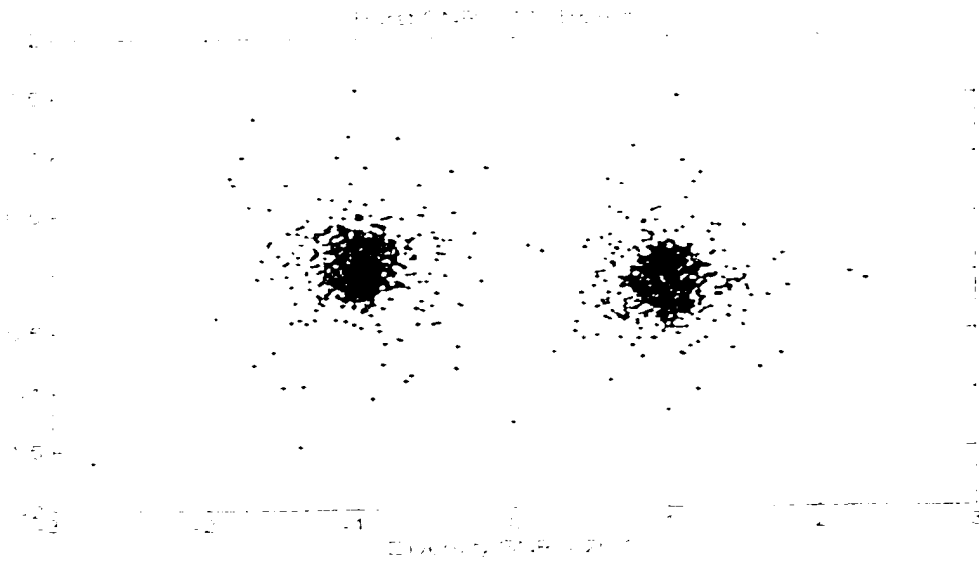


圖 B-11：在第二次疊代之後，限幅器輸入，MoCA10408 頻道，15[dB]AWGN SNR



在第三探測 II 處理之後，SNR 大約為 10.6[dB]

圖 B-12：在第三次疊代之後，頻率圖，MoCA10408 通道，15[dB]AWGN SNR

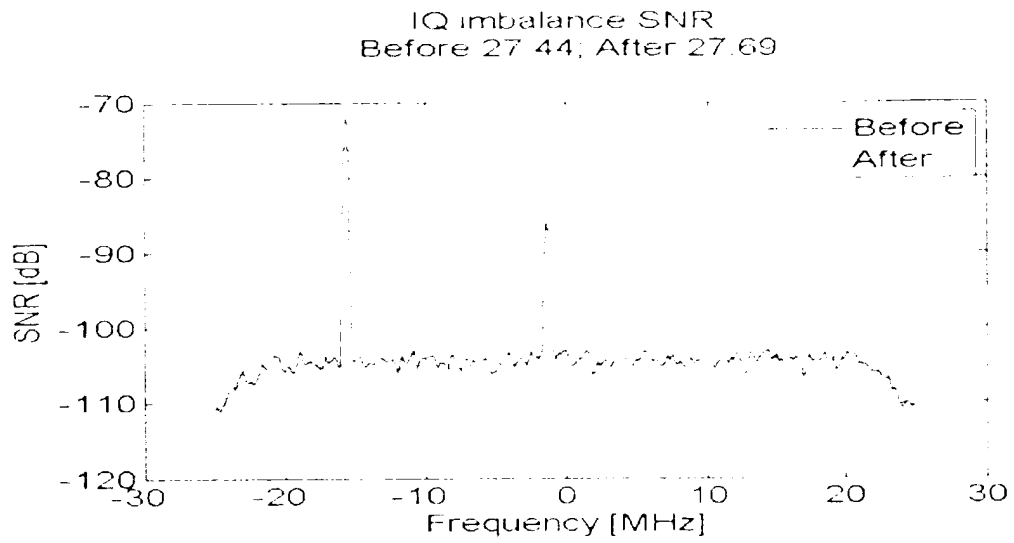


圖 B-13：在第三次疊代之後，限幅器輸入，MoCA10408 通道，15[dB]AWGN SNR

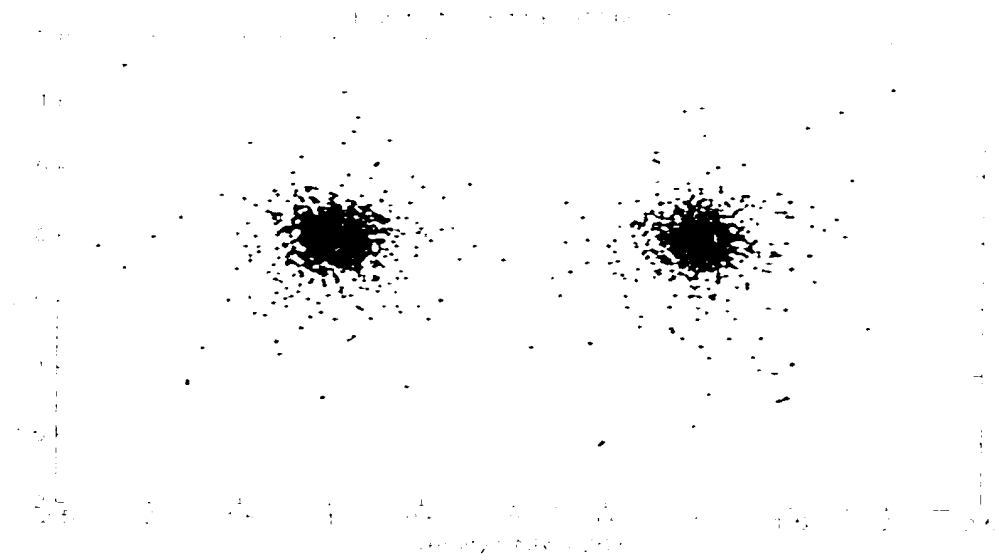
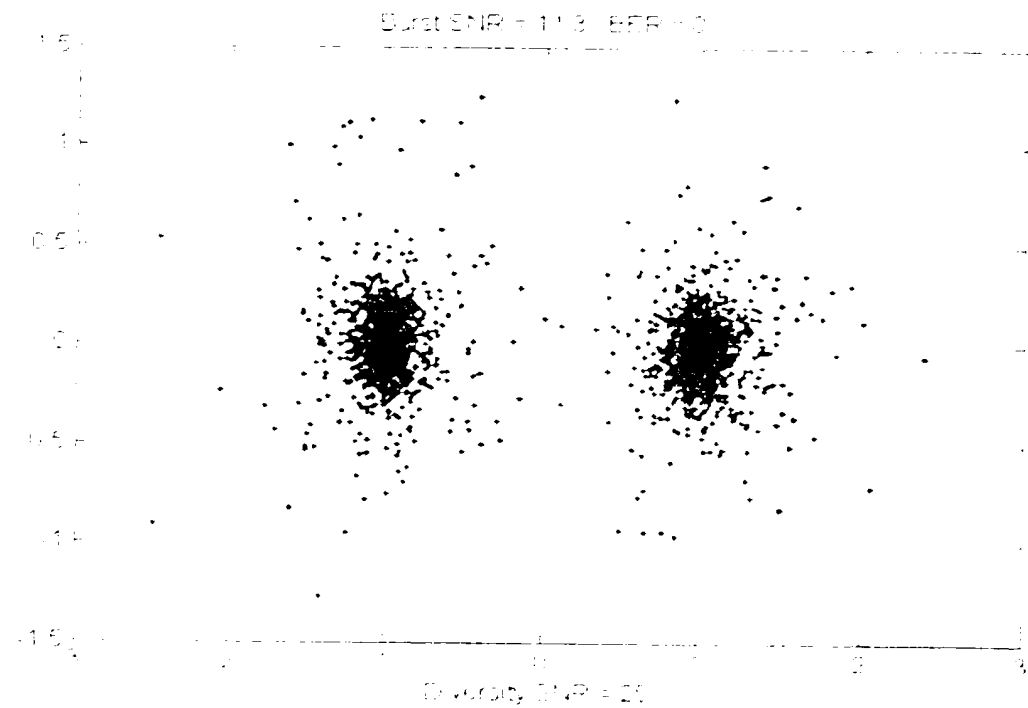


圖 B-14 顯示，I/Q 平衡，SNR 約為 11.3 [dB]，因此，發射 I/Q 失衡對於解調器的不亂噪音，並且即使在哈布(charsh)補遺條件下，估計和補償演算法都是有用的。

圖 B-14: 限幅器輸入，MoCA10408 通道，15 [dB]AWGN SNR，沒有 IQ 失衡



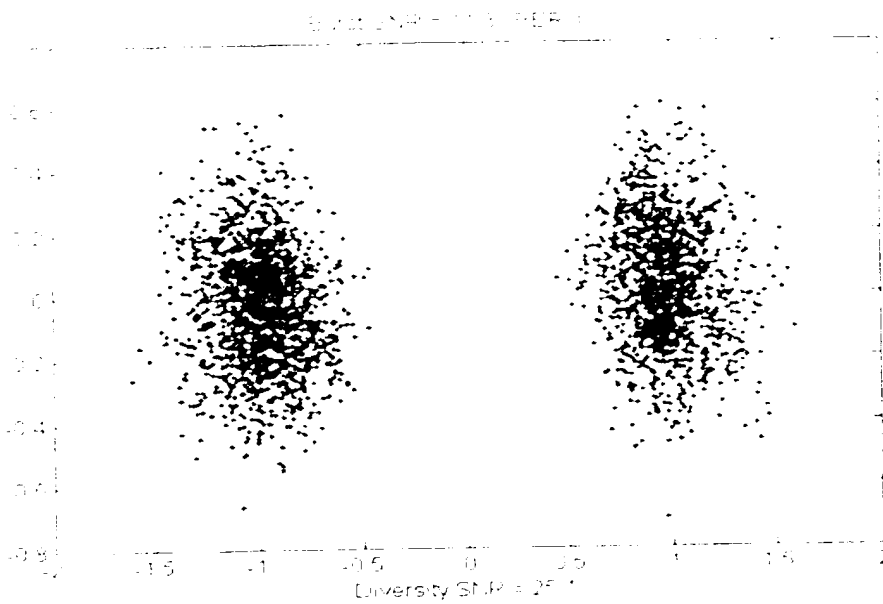
TX 與 RX IQ 平衡的誤差

MoCA 實現，對於接收機接收者，TX 與 RX IQ 失衡在對的

再將，為了點，我們知道，對於 TX 失調，其的電流，以及 RX 參數，我們知道結果如下：

- TX 振幅失調 1[dB]，TX 相位失調 2°
- RX 振幅失調 3[dB]
- RX 相位失調 10°，輸出偏移 20ppm
- 信道：理想、沒有 AWGN

圖 B-15：在 TX 和 RX I/Q 失調下，基帶接收器、理想信道、沒有 AWGN



第一例探測 2 傳輸後，SNR 大約為 21.2[dB]

圖 B-16：在 TX 和 RX IQ 失調下，第二例迭代後，頻帶圖、理想信道、沒有 AWGN

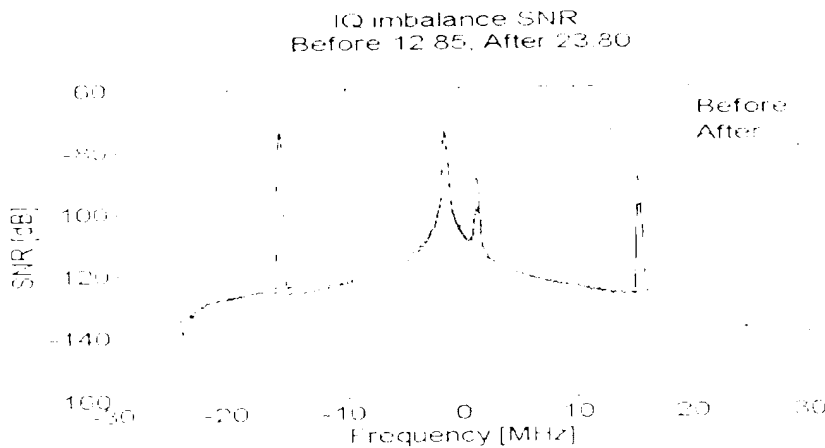
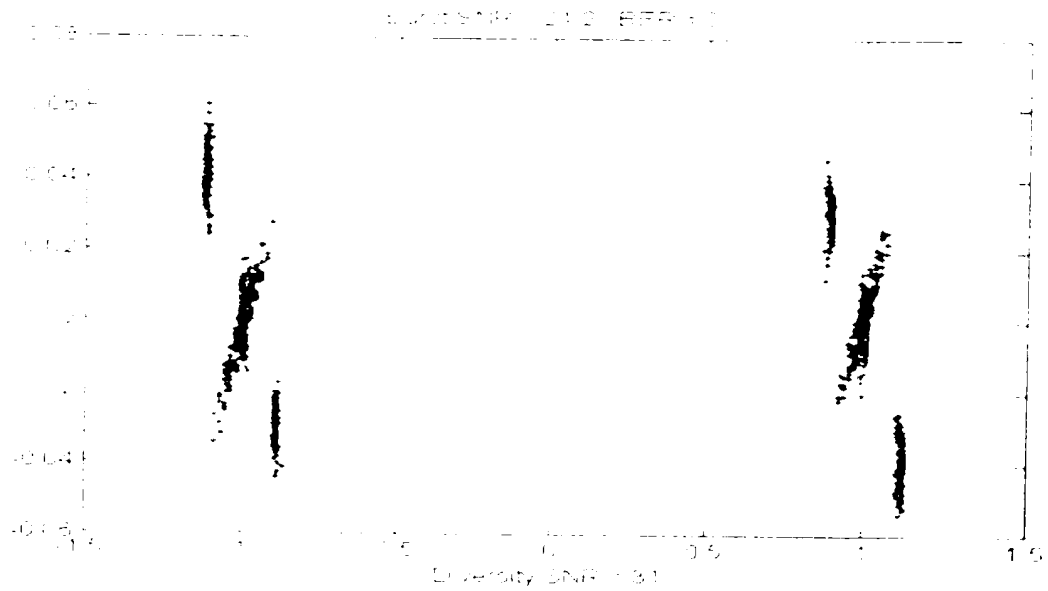


圖 B-17：在 TX IQ 失衡下，經過 3 次疊代後，限幅器輸入、理想通道、沒有 AWGN

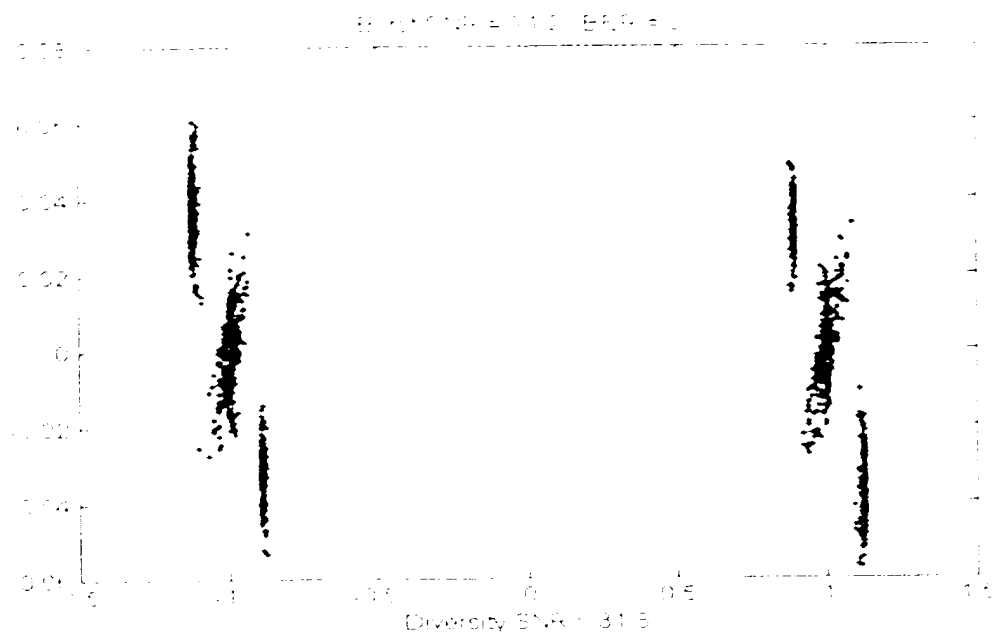


在 3 次疊代後，所估計的 RX I/Q 失衡參數是

參數	實值	估計
$G$	0.7071	0.70785
$\Theta$	10.0	10.2014

因此，對參數進行了準確的估計，與說明書中的 SMR 相比，其中帶 TX 失衡參數，大約為 21.2[dB]

圖 B-18：在僅僅 TX IQ 失衡下，限幅器輸入、理想通道、沒有 AWGN



因此，在 TXI/Q 失靈出現的情況下，所提出的演算法是有用的。

### 定點虛擬碼

如下的虛擬碼給出了上述演算法的定點實現。注意複變數由字母“c”表示。

```

Function 1: Probe2Processing
function [rho, theta, Scale, Q] = Probe2Processing (CFO,
clQparameters_log,i1,i2, N_delta)
if (CFO==0)
    CFO=-1;
end
% Residual Frequency Estimation
[phasor_p,phasor_m] =
Residual_Frequency_Estimation(clQparameters_log,CFO);

```

```
% Residual Frequency Correction
```

```
cZ =
```

```
Residual_frequency_Correction(cIQparameters_log,cphasor_p,cphasor_m);
```

```
% Coefficient Computation
```

```
[cfB1,cfB2,Scale_ratio] = Coeff_Computation(CFO,i1,i2);
```

```
% Phasor Estimation
```

```
[cg_exp_mTeta_M_16,Scale_g,scale_inv] = Phasor_Estimation(cZ,cfB1,cfB2,CFO,N_delta,Scale_ratio);
```

```
% IQ Coeff Computation
```

```
[theta,rho, Scale_Q] =
```

```
Compensation_Params_Estimation(cg_exp_mTeta_M_16,scale_inv,Scale_g);
```

表 C-1：探測 2 處理變數定義表

名稱	大小	解釋	I/O
CFO	32bit	載波頻率偏移	I
cIQparameters_log	16bit x 6 x Nsym Complex	記錄 FFT 輸出	I
I1	8bit	負窗口偏移 1	I
I2	8bit	負窗口偏移 2	I
N_delta	16bit	NCO 重定時間	I
cphasor_p	16bit Complex	頻率補償	NA
cphasor_m	16bit Complex	頻率補償	NA
cZ	16bit x 6 Complex	旋轉 FFT 輸出	NA
cfB1	16bit Complex	估計係數	NA
cfB2	16bit Complex	估計係數	NA
Scale_ratio	16bit	縮放 (Scaling) 因數	NA
Cg_exp_mTeta_M_16	16bit Complex	經估計 I/Q 向 量	NA
Theta	16bit	I/Q 補償	O
Rho	16bit	I/Q 補償	O

```

Function 2: Residual_Frequency_Estimation
function [cphasor_p,cphasor_m] =
Residual_Frequency_Estimation(cIQparameters_log,CFO)

%TBD Select 1 or 2 according to SNR
cfphasor_64 = 0;

for i=0:Nsym-2
    cfphasor_64 = Cmplx_Add_64_32(cfphasor_64,
Cmplx_Mult_16_16(cIQparameters_log(i,1),...
conj(cIQparameters_log(i+1,1)));
end

%Level control to 16 bit signed
Nphasor_bits = Nfft_out-1;

csphasor = Scale_Complex_64(cfphasor_64, Nphasor_bits);

%Get phasor angle and magnitude
[angle_Ef rPhasor] = cordic_SW( csphasor,1);

%Generate 'Exp_Vec_p'
fScale = 26981;                                %const 16 bit -
round(gcordic)*2^(Nfft_out-1))

csphasor_div=Cmplx_real_div_32_16 (csphasor<<(Nfft_out-2)),
rPhasor); %32bit complex / 16bit real division
% csphasor_div <+-2^15

```

```

cphasor_p_32 =
(Cmplx_real_mult_16_16(fScale,cphasor_div) )>>(Nfft_out-2);
%Scale back to 16bit, known

cphasor_p = Cmplx_Saturate(cphasor_p_32, Nphasor_bits);

%Compute angle for IQ image rotation (Coridic Preparations)
angle_m = 2*CFO*(Nfft+LCP) - angle_Ef;          %1rad=
2^(Fr_bits-1)

%Generate 'Exp_Vec_m'
[cphasor_m tmp] = cordic_SW( angle_m,0); %gives -phasor(angle)

Function 3: Scale_Complex_64
function [cphasor] = Scale_Complex_64(cfphasor_64, Nphasor_bits)

Ceil_Log2_Abs_Real_Cfphasor_64 =
ceil_log2(abs(real(cfphasor_64)));
Ceil_Log2_Abs_Imag_Cfphasor_64 =
ceil_log2(abs(imag(cfphasor_64)));
Scale=Nphasor_bits-max(Ceil_Log2_Abs_Real_Cfphasor_64,Ceil_Log2_Abs_Imag_Cfphasor_64);
if (Scale>=0)
cphasor_32 = (cfphasor_64<< Scale);
else
cphasor_32 = (cfphasor_64>> (-Scale));
end
cphasor = Cmplx_Saturate(cphasor_32, Nphasor_bits);

```

Function 4: ceil\_log2

```
function [i] = ceil_log2 (X)
```

```
    i=0;
```

```
    while(X!=0)
```

```
        X=X>>1;
```

```
        i=i+1;
```

```
    end
```

Function: 5: Sign

```
function [Y] = Sign (X)
```

```
    y=1;
```

```
    if (X<0)
```

```
        y=-1;
```

```
    end
```

Function 6: Cmplx\_Saturate

```
function [X_16] = Cmplx_Saturate(X_32,Nbits)
```

```
    Sign_Real_X_32 = sign(real(X_32));
```

```
    Sign_Imag_X_32 = sign(imag(X_32));
```

```
    Abs_Real_X_32 = abs(real(X_32));
```

```
    Abs_Imag_X_32 = abs(imag(X_32));
```

```
    if (Abs_Real_X_32 >= (1<<Nbits) )
```

```
        if(Sign_Real_X_32==1)
```

```
            X_32.r= (1<<Nbits)-1;
```

```
        else
```

```

X_32.r= -( (1<< Nbits)-1 );
    end
end

if(Abs_Imag_X_32 >= (1<< Nbits) )
    if(Sign_Image_X_32==1)
        X_32.i= (1<< Nbits)-1;
    else
X_32.i= -( (1<< Nbits)-1 );
    end
end
end
X_16 = X_32;    %casting to 16bit

```

表 C-2 : Residual\_Frequency\_Estimation, Scale\_Complex\_64 變數定義表

名稱	大小	解釋	I/O
cfphasor64	64 bit complex	向量 Acc	NA
<u>Nphasor_bits</u>	8bit	Target Num of phasor bits	NA
csphasor_32	32bit complex	Tmp Variable	NA
csphasor	16bit complex	Scaled phasor for freq rot	NA
rPhasor	16bit	大小 csphasor	NA
csphasor_div	16bit complex	標準化向量	NA
cphasor_p	16bit complex	頻率補償	O
angle_Ef	32bit	Csphasor 的角度	NA
angle_m	32bit	phasor_m rotation 的角度	NA
cphasor_m	16bit complex	頻率補償	O

表 C-3： SW CORDIC 變數定義

名稱	大小	解釋	I/O
x_tmp	32bit x 2 array	本地存儲 (使用不大於 17 bits)	NA
Atan_Table	16bit x13 array	ATan Table (const)	NA
data_in	32bit complex	CORDIC 輸入	I
Angle	32bit	向量角	NA
Angle_offset	32bit	向量角偏移	NA
X	32bit x 2 array	CORDIC 輸出 1	O
z	32bit x 2 array	CORDIC 輸出 2	O
mode	Boolean (1bit)	CORDIC 模式選擇	I

## Function 7: Cordic\_Pre\_Process

```
function [data_in, X] = Cordic_Pre_Process(data_in,mode)
```

```
%*****
```

```
*****
```

```
% Pre Cordic Processing Bring angle to [-pi/2,pi/2] and find  
Quadrant
```

```
%*****
```

```
*****
```

```
if(mode==0)
```

```
    Npi = 0;
```

```
    Sgn=0;
```

```
    while (data_in < -QUARTER)
```

```
        data_in = data_in + (QUARTER<<1);
```

```
        Npi = 1-Npi;
    end

    while (data_in > QUARTER)
        data_in = data_in - (QUARTER<<1);
        Npi=1-Npi;
    end

    X(0) = 19898;
    % (1/GainCordic)*2^(Ntan-1)
    X(1) = 0;

    if(Npi)
        X=-X;
    end
else
    X=0;
    Sgn_real = sign(real(data_in));
    Sgn_imag = sign(imag(data_in));

    if ( Sgn_real ==-1)
        data_in = -data_in;
        if(Sgn_imag == 1)
            X = QUARTER<<1;
        else
            X = -(QUARTER<<1);
        end
    end
end
end
```

表 C-4: Cordic\_SW, Cordic\_Pre\_Process 變數定義

名稱	大小	解釋	I/O
data_in	32bit complex	CORDIC 輸入	I/O
X	16bit x 2 array	CORDIC 向量	O

## Function 8: Residual\_frequency\_Compensation

```
function [cZ_16] =
```

```
Residual_frequency_Correction(cIQparameters_log,cphasor_p,cphasor_m)
```

```
cdphasor_p = cphasor_p;
```

```
cdphasor_m = cphasor_m;
```

```
cZ_ACC_64 = zeros(1,6);%64bit Acc array (16bit phasor * 16bit FFT output + log2(40)bit for acc >32 !!!!)
```

```
for i=1:Nsym-1
```

```
    cZ_ACC_64(1) = Cmplx_Add_64_32(cZ_ACC_64(1),  
    Cmplx_Mult_16_16(cIQparameters_log(i+1,1)  
    ,cphasor_p);
```

```
    cZ_ACC_64(2) = Cmplx_Add_64_32(cZ_ACC_64(2),  
    Cmplx_Mult_16_16(cIQparameters_log(i+1,2)  
    ,cphasor_p);
```

```
    cZ_ACC_64(3) = Cmplx_Add_64_32(cZ_ACC_64(3),  
    Cmplx_Mult_16_16(cIQparameters_log(i+1,3)  
    ,cphasor_m);
```

```

    cZ_ACC_64(4) = Cmplx_Add_64_32(cZ_ACC_64(4),
Cmplx_Mult_16_16(cIQparameters_log(i+1,4)
    ,cphasor_m);
    cZ_ACC_64(5) = Cmplx_Add_64_32(cZ_ACC_64(5),
Cmplx_Mult_16_16(cIQparameters_log(i+1,5)
    ,cphasor_m);
    cZ_ACC_64(6) = Cmplx_Add_64_32(cZ_ACC_64(6),
Cmplx_Mult_16_16(cIQparameters_log(i+1,6)
    ,cphasor_m);
    cphasor_p = Cmplx_Saturate ((Cmplx_Mult_16_16
(cphasor_p,cdphasor_p))>>(Ntan-1), Ntan-1);
    cphasor_m = Cmplx_Saturate( (Cmplx_Mult_16_16
(cphasor_m.cdphasor_m))>>(Ntan-1), Ntan-1);

end

%Scale Back to fit 32bit
for i=1:6
cZ_ACC_32(i) = cZ_ACC_64(i)>>(Nfft_out);
end
% Level control vector so that max fits in 16bit

Max_Z=0;
for i=1:6
    if (abs(real(cZ_ACC_32(i)))>Max_Z) .
        Max_Z = abs(real(Z_ACC_32(i)));
    end
    if (abs(imag(cZ_ACC_32(i)))>Max_Z)
        Max_Z = abs(imag(Z_ACC_32(i)));
    end
end

```

```

end
%Scale Back to 16 bits
Scale_Z = (Nfft_out-1) - ceil_log2(Max_Z);

for i=1:6
cZ_16(i) = Cmplx_Saturate (cZ_ACC_32(i)>>(Scale_Z), Nfft_out-1);
end

```

表 C-5: Residual\_frequency\_Compensation 變數

名稱	大小	解釋	I/O
cdphasor_p	16bit Complex		NA
cdphasor_m	16bit Complex		NA
cphasor_p	16bit Complex		I
cphasor_m	16bit Complex		I
cZ_ACC_64	64bit Complex x 6 array		NA
cZ_32	32bit Complex x 6 array		NA
cZ_16	16bit Complex x 6 array		O
Max_32	32bit		NA
ScaleZ	16bit		NA

Function: 9: Coeff Computation

```
function [cfB1,cfB2,Scale_ratio] = Coeff_Computation(CFO,i1,i2)
```

```

Log2Nfft = 8;           %log2(Nfft) = log2(256)=8;
PI_over_FFT = 804;     % round(pi/Nfft*2^(Fr_bits-1))
Nfft=256;
N1=2;

```

```
[cphasor11 tmp] = cordic_SW( (CFO<<(1+Log2Nfft)),0);

cphasor11 = Switch_real_imag(cphasor11);
cfs11 = Cmplx_Add_16_16 (phasor11, -sqrt(-1)*(1<<(Ntan-1)));

fs21 = CFO+ PI_over_FFT *i1;
fs22 = CFO +PI_over_FFT *i2;

if(fs21==0)
    cfB1_32 = 1<<( Nfft_out+ Log2Nfft );
else
    cfB1_32 = Cmplx_real_div_32_16
((cfs11<<(Nfft_out-N1)),fs21);
end

if(fs22==0)
    cfB2_32 = 1<<( Nfft_out+ Log2Nfft );
else
    cfB2_32 = Cmplx_real_div_32_16
((cfs11<<(Nfft_out-N1)),fs22);
end

%Scale to fB1,fB2 to 15bit signed
Max =0;
Max_abs_real_cfB1 = abs(real(cfB1_32));
Max_abs_imag_cfB1= abs(imag(cfB1_32));
Max_abs_real_cfB2 = abs(real(cfB2_32));
Max_abs_imag_cfB2= abs(imag(cfB2_32));

if(Max_abs_real_cfB1 >Max)
    Max = Max_abs_real_cfB1;
```

```

end
if(abs Max_abs_imag_cfB1>Max)
    Max = Max_abs_imag_cfB1;
end
if(Max_abs_real_cfB2 >Max)
    Max = Max_abs_real_cfB2
end
if(Max_abs_imag_cfB2>Max)
    Max = Max_abs_imag_cfB2;
end

Scale_ratio = 15-(ceil_log2(Max)+1);
If (Scale>=0)
    cfB1 = Cmplx_Saturate( (cfB1_32<<Scale_ratio), Nfft_out-2);
    cfB2 = Cmplx_Saturate ((cfB2_32<<Scale_ratio),Nfft_out-2);
else
    cfB1 = Cmplx_Saturate ((cfB1_32>>-Scale_ratio), Nfft_out-2);
    cfB2 = Cmplx_Saturate ((cfB2_32>>-Scale_ratio), Nfft_out-2);
end

```

表 C-6：係數計算變數定義

名稱	大小	解釋	I/O
cphasor11	16bit complex	CORDIC 輸出	NA
cfs11	32bit complex	係數分子	NA
fs21	16bit	係數分母	NA
fs22	16bit	係數分母	NA
cfB1_32	32bit complex	非縮放 1	NA
cfB2_32	32bit complex	非縮放 2	NA
cfB1	16bit complex	係數 1	O
cfB2	16bit complex	係數 2	O
Scale_ratio	16bit	係數縮放因數	O

## Function 10: Phasor Estimation variable definition

```

function [cg_exp_mTeta_M_16,Scale_g,scale_inv] =
Phasor_Estimation(cZ,cfB1,cfB2,CFO,N_delta,Scale_ratio)

N2=2;

%*****
%*****
%   Fix Point Computation of
%
%   Z1*(|fB1|^2+|fB2|^2)      Z1*(|fB1|^2+|fB2|^2)*(B1'*Z3
+B2*Z4)'
% ----- = -----
%   B1'*Z3 +B2*Z4'
|B1'*Z3|^2 +|B2*Z4|^2
%*****
%*****
sumfB1SfB2S = (Real_Add_30_30_T(MAG_2_16(cfB1),
MAG_2_16(cfB2)))>>15

cNUMERATOR_32(0) =
Cmplx_Real_Mul_16_16(cZ(0),sumfB1SfB2S);
cNUMERATOR_32(1) =
Cmplx_Real_Mul_16_16(cZ(1),sumfB1SfB2S);

cDENOM_32(0)
=Cmplx_Add_32_32(Cmplx_Mul_16_16(conj(cZ(2)),(cfB1)),
Cmplx_Mul_16_16(conj(cZ(3)),(cfB2)));

```

```
cDENOM_32(1) =  
Cmplx_Add_32_32(Cmplx_Mul_16_16(conj(cZ(4)),(cfB1)),  
  
Cmplx_Mul_16_16(conj(cZ(5)),(cfB2)));  
  
cMul_NUM_cDENOM_64(0) =  
Cmplx_Mul_32_32(cNUMERATOR_32(0),conj(cDENOM_32(0)));  
cMul_NUM_cDENOM_64(1) =  
Cmplx_Mul_32_32(cNUMERATOR_32(1),conj(cDENOM_32(1)));  
  
DENOM_2_64(0) = MAG_2_32(cDENOM_32(0));  
DENOM_2_64(1) = MAG_2_32(cDENOM_32(1));  
%Level Control for division scale denominator down to 32bit  
if(DENOM_2_64(0)>DENOM_2_64(1) )  
  
    Scale = ceil_log2(DENOM_2_64(0));  
else  
    Scale = ceil_log2(DENOM_2_64(1));  
end  
  
Scale = 31-Scale;  
  
If(Scale>=0)  
DENOM_2_32(0) = Cmplx_Saturate  
(DENOM_2_64(0)<<(Scale),31);  
DENOM_2_32(1) = Cmplx_Saturate (DENOM_2_64(1)<<  
(Scale),31);  
else  
DENOM_2_32(0) =
```

```

Cmplx_Saturate( DENOM_2_64(0)>>(-Scale),31);
DENOM_2_32(1) = Cmplx_Saturate
(DENOM_2_64(1)>>(-Scale),31);
end

```

```

%complex/real division 64bit/32bit gives 32bit result

```

```

cfCM(0) = Cmplx_real_div_64_32
(cMul_NUM_cDENOM_64(0)>>N2 ,DENOM_2_32(0));
cfCM(1) = Cmplx_real_div_64_32
(cMul_NUM_cDENOM_64(1)>>N2 ,DENOM_2_32(1));

```

```

cfCM_avg = Cmplx_Add_32_32 (cfCM(0),cfCM(1));

```

```

% Rotation of FCM_avg needed only if HW can not Insure Delatn=0

```

```

[cphasor_dn tmp] = cordic_SW( (CFO<<1)*N_delta, 0);

```

```

cfCM_avg_rot_64 = Cmplx_Mul_32_16(cfCM_avg,cphasor_dn);

```

```

cfCM_avg_rot_32 = fCM_avg_rot_64>>(Ntan);

```

```

Scale_fCM = Ntan-1-N1+Scale_ratio-Scale-6-N2;

```

```

fCM_one_level_32 = 1<<(Scale_fCM);

```

```

cfp_m_32 = Cmplx_Real_Add_32_32( cfCM_avg_rot_32,
-fCM_one_level_32);

```

```

cfp_p_32 = Cmplx_Real_Add_32_32( cfCM_avg_rot_32,
fCM_one_level_32);

```

```

cNUMERATORf_64 =

```

```

Cmplx_Mul_32_32(cfp_m_32,conj(cfp_p_32));

```

```

DENOMf_64 = MAG_2_32 (cfp_p_32);

```

%Level Control for division scale denominator down to 32bit signed

```
Abs_real_Numerator64 = abs(real(cNUMERATORf_64));
Abs_imag_Numerator64 = abs(imag(cNUMERATORf_64));

if(Abs_real_Numerator64 > Abs_imag_Numerator64)
    Scale_fn = 30 - ceil_log2(Abs_real_Numerator64);
else
    Scale_fn = 30 - ceil_log2(Abs_imag_Numerator64);
end

if(Scale_fn >= 0)
    cNUMERATORf_32 = cNUMERATORf_64 << (Scale_fn);
```

```
else
    cNUMERATORf_32 = cNUMERATORf_64 >> (-Scale_fn);
end
```

%Level Control for division scale denominator down to 16bit

unsigned

```
Scale_fd = 16 - ceil_log2(DENOMf_64);
```

```
if(Scale_fd >= 0)
    DENOMf_scaled_16 = Cmplx_Saturate
    (DENOMf_64 << (Scale_fd), 16);
else
    DENOMf_scaled_16 = Cmplx_Saturate
    (DENOMf_64 >> (-Scale_fd), 16);
end
```

% Division 32bit complex by 16bit real

```
cg_exp_mTeta_M_16 = Cmplx_real_div_32_16
(cNUMERATORf_32,DENOMf_scaled_16);
```

```
Scale_g = Scale_fd-Scale_fn;
scale_inv = 14-Scale_fd+Scale_fn;
```

表 C-7：向量估計變數定義

名稱	大小	解釋	I/O
sumfB1SfB2S	16bit		
cNUMERATOR_32	32bit x 2 complex array		
cDENOM_32	32bit x 2 complex array		
cMul_NUM_cDENOM_64	64bit x 2 complex array		
DENOM_2_64	64bit x 2 array		
DENOM_2_32	32bit x 2 array		
cfCM	32bit x 2 complex array		
cfCM_avg	32bit complex		
cphasor_dn	32bit complex		
cfs11	16bit complex		
fs21	16bit		
fs22	16bit		
cfCM_avg_rot_64	64bit complex		
cfCM_avg_rot_32	32bit complex		
cfCM_one_level_32	32bit		
cfp_m_32	32bit complex		
cfp_p_32	32bit complex		
cNUMERATORf_64	64bit complex		
DENOMf_64	64bit		
Abs_real_Numerator64	64bit		
Abs_imag_Numerator64	64bit		
cNUMERATORf_32	32bit complex		
DENOMf_scaled_16	16bit complex		
cg_exp_mTeta_M_16	16bit complex		O
Scale_g	16bit		O
scale_inv	16bit		O
Scale_fd	16bit		
Scale_fn	16bit		
Scale_fCM	16bit		
Scale	16bit		

Function 11: Compensation\_Params\_Estimation

function [theta,rho, Scale\_Q] =

Compensation\_Params\_Estimation(cg\_exp\_mTeta\_M\_16,  
scale\_inv,Scale\_g)

Mue\_bits=4;

if(ProbeII\_Num == 1)

Mue =12;

elseif(ProbeII\_Num == 2)

Mue = 8;

else

Mue = 4;

End

if(ProbeII\_Num>=1)

% First Order Loop

Mue\_1m = (16-Mue);

cg\_ACC\_32 = cg\_exp\_mteta <<(-Scale\_g-Mue\_bits);

cg\_Delta\_32 = (Cmplx\_Mul\_16\_16 (cg\_exp\_mteta  
\*cg\_exp\_mTeta\_M\_16))>>4;

cg\_ACC\_32 = Cmplx\_Add\_32\_32

(Cmplx\_Mul\_32\_16(cg\_ACC\_32,Mue\_Fix\_1m),

Cmplx\_Mul\_32\_16(cg\_Delta\_32,Mue\_Fix\_1m))>>(-Scale\_g);

cg\_exp\_mteta = g\_ACC\_32;

% I/Q Correction Params Calculation

[theta,rho, Scale\_Q] =

Compute\_Fix\_Point\_IQ\_Coeffs(cg\_exp\_mteta,scale\_inv,Scale\_g,0);

else

```

g_exp_mteta = cg_exp_mTeta_M_16;
%      I/Q Correction Params Calculation
[theta,rho, Scale_Q] =
Compute_Fix_Point_IQ_Coeffs(g_exp_mTeta_M_16,scale_inv,
Scale_g,1);
end
ProbeII_Num = ProbeII_Num+1;

```

表 C-8: Compensation\_Params\_Estimation 變數定義

名稱	大小	解釋	I/O
ProbeII_Num	16bit	重定值為 0	NA
Mue	16bit	環增益	NA
Mue_1m	16bit	補充環增益	NA
cg_ACC_32	32bit complex		
cg_Delta_32	32bit complex		
cg_exp_mteta	16bit complex		
theta	16bit	重定值 2048	O
rho	16bit	重定值 0	O
Scale_Q	1bit	重定值 1	O

Function 13: Compute\_Fix\_Point\_IQ\_Coeffs

```

function [theta,rho, Scale_Q] =
Compute_Fix_Point_IQ_Coeffs(cg_exp_mTeta_M_16,scale_inv,
scale_g,FirstTime)
if(FirstTime)
    scale_delta=0;

```

```
    if(scale_inv>31)
        scale_delta = scale_inv-31;
        scale_inv=31;
    end
else

    scale_inv = scale_inv_log;
    scale_delta= scale_delta_log;
    scale_g = scale_g_log;
end

real_g_exp_mTeta_M_16 = real(cg_exp_mTeta_M_16);
imag_g_exp_mTeta_M_16 = imag(cg_exp_mTeta_M_16);

if( real_g_exp_mTeta_M_16 > 1<< (-scale_g_log) )

    Inv_real_g_exp_mTeta_M_16 = Cmplx_real_div_32_16
(1<<(scale_inv),real_g_exp_mTeta_M_16));
    Scale_theta = -14+Teta_bits-1+scale_delta;

    If(Scale_theta>=0)
theta = Inv_real_g_exp_mTeta_M_16<<(Scale_theta);
    else
theta = Inv_real_g_exp_mTeta_M_16>>(-Scale_theta);
    end
```

```
Rho_32 = Cmplx_Mul_16_16
(-imag_g_exp_mTeta_M_16,Inv_real_g_exp_mTeta_M_16);
Scale_rho = -Rho_bits+1+scale_inv;
If(Scale_rho>=0)
rho = Rho_32<<(Scale_rho);
else
rho = Rho_32>>(-Scale_rho);
end

Scale_Q = 1;
else
Scale_theta = scale_g_log +Teta_bits-1;
Scale_rho = scale_g_log + Rho_bits -1;

If(Scale_theta>=0)
theta = real_g_exp_mTeta_M_16<<(Scale_theta);
else
theta = real_g_exp_mTeta_M_16>>(-Scale_theta);
end

If(Scale_rho>=0)
rho = (-imag_g_exp_mTeta_M_16)<<(Scale_rho);
else
rho = (-imag_g_exp_mTeta_M_16)>>(-Scale_rho);
end

Scale_Q = 0;
end
```

表 C-9: Compute\_Fix\_Point\_IQ\_Coeffs 變數定義

名稱	大小	解釋	I/O
scale_inv_log	16bit		
scale_delta_log	16bit		
scale_inv	16bit		
scale_delta	16bit		
real_g_exp_mTeta_M_16	16bit		
imag_g_exp_mTeta_M_16	16bit		
Inv_real_g_exp_mTeta_M_16	16bit		
Scale_theta	16bit		
Scale_rho	16bit		
Rho_32	32bit		
theta	16bit		O
rho	16bit		O
Scale_Q	1bit		O

## 複變數數學運算操作定義

$$[C_{32_r}, C_{32_i}] =$$

$$\text{Cmplx\_Add}_{16\_16}(A_{16_r}, A_{16_i}, B_{16_r}, B_{16_i});$$

$$C_{32_r} = A_{16_r} + B_{16_r}$$

$$C_{32_i} = A_{16_i} + B_{16_i}$$

$$[C_{32_r}, C_{32_i}] =$$

$$\text{Cmplx\_Add}_{32\_32}(A_{32_r}, A_{32_i}, B_{32_r}, B_{32_i});$$

$$C\_32\_r = A\_32\_r + B\_32\_r$$

$$C\_32\_i = A\_32\_i + B\_32\_i$$

$$[R\_32] = \text{Real\_Add\_30\_30\_T}(A\_30, B\_30);$$

$$R\_32 = A\_30 + B\_30$$

$$[C\_64\_r, C\_64\_i] =$$

$$\text{Cmplx\_Add\_64\_32}(A\_64\_r, A\_64\_i, B\_32\_r, B\_32\_i);$$

$$C\_64\_r = A\_64\_r + B\_32\_r$$

$$C\_64\_i = A\_64\_i + B\_32\_i$$

$$[C\_32\_r, C\_32\_i] = \text{Cmplx\_Real\_Add\_32\_16}(A\_32\_r, A\_32\_i, B\_16);$$

$$C\_32\_r = A\_32\_r + B\_16$$

$$C\_32\_i = A\_32\_i$$

$$[C\_32\_r, C\_32\_i] = \text{Cmplx\_imag\_Add\_32\_16}$$

$$(A\_32\_r, A\_32\_i, B\_16);$$

$$C\_32\_r = A\_32\_r$$

$$C\_32\_i = A\_32\_i + B\_16$$

$$[C\_32\_r] = \text{MAG\_2\_16}(A\_16\_r, A\_16\_i)$$

$$C\_32\_r = A\_16\_r * A\_16\_r + A\_16\_i * A\_16\_i$$

$$[C\_64\_r] = \text{MAG\_2\_32}(A\_32\_r, A\_32\_i)$$

$$C\_64\_r = A\_32\_r * A\_32\_r + A\_32\_i * A\_32\_i$$

$$[C\_32\_r, C\_32\_i] =$$

$$\text{Cmplx\_Mult\_16\_16}(A\_16\_r, A\_16\_i, B\_16\_r, B\_16\_i)$$

$$C\_32\_r = A\_16\_r * B\_16\_r - A\_16\_i * B\_16\_i$$

$$C\_32\_i = A\_16\_r * B\_16\_i + A\_16\_i * B\_16\_r$$

$$\begin{aligned}
 [C_{64_r}, C_{64_i}] &= \\
 &\text{Cmplx\_Mult}_{32\_32}(A_{32_r}, A_{32_i}, B_{32_r}, B_{32_i}) \\
 C_{64_r} &= A_{32_r} * B_{32_r} - A_{32_i} * B_{32_i} \\
 C_{64_i} &= A_{32_r} * B_{32_i} + A_{32_i} * B_{32_r}
 \end{aligned}$$

$$\begin{aligned}
 [C_{32_r}, C_{32_i}] &= \text{Cmplx\_real\_Mult}_{16\_16} \\
 &(A_{16_r}, A_{16_i}, B_{16_r}) \\
 C_{32_r} &= A_{16_r} * B_{16_r} \\
 C_{32_i} &= A_{16_i} * B_{16_r}
 \end{aligned}$$

$$\begin{aligned}
 [C_{64_r}, C_{64_i}] &= \text{Cmplx\_Mul}_{32\_16}(A_{32_r}, A_{32_i}, B_{16_r}) \\
 C_{64_r} &= A_{32_r} * B_{16_r} \\
 C_{64_i} &= A_{32_i} * B_{16_r}
 \end{aligned}$$

$$\begin{aligned}
 [C_{16_r}, C_{16_i}] &= \text{Cmplx\_real\_div}_{32\_16}(A_{32_r}, A_{32_i}, B_{16_r}) \\
 C_{32_r} &= A_{32_r} / B_{16_r} \\
 C_{32_i} &= A_{32_i} / B_{16_r}
 \end{aligned}$$

$$\begin{aligned}
 [C_{32_r}, C_{32_i}] &= \text{Cmplx\_real\_div}_{64\_32}(A_{64_r}, A_{64_i}, B_{32_r}) \\
 C_{32_r} &= A_{64_r} / B_{32_r} \\
 C_{32_i} &= A_{64_i} / B_{32_r}
 \end{aligned}$$

$$\begin{aligned}
 [C_{16_r}, C_{16_i}] &= \text{Switch\_real\_imag}(A_{16_r}, A_{16_i}); \\
 C_{32_r} &= A_{16_i}
 \end{aligned}$$

$$C_{32_i} = A_{16_r}$$

## 附錄 D

### 示範性的 HW-SW 介面參數

下表總結了在探測 2 接收的過程中，HW 和 SW 之間的資訊交換。輸出指 HW 的輸出，輸入指輸入到 HW。

HW 名稱	SW 名稱	大小	輸入到/輸出自 HW	備註
$\Delta n$	N_delta	16bits	輸出	NCO 重定和 FFT 窗口的第一採樣點之間的採樣點數量
i1	I1	3 bit	輸出	名義上的第一負數 FFT 窗口的偏移系
i2	I2	4 bit	輸出	名義上的第二負數 FFT 窗口的偏移系
CFO	CFO	17bit	輸出	載波頻率偏移 (在精確的頻率偏移後)
Z[k1,:]	cIQparameters_log(0,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 k1 for symbols 1,..., Nprobe2_symbols
Z[k2,:]	cIQparameters_log(1,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 k2 for symbols 1,..., Nprobe2_symbols
Z[-k1+i1,:]	cIQparameters_log(2,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 -k1+i1 for symbols 1,..., Nprobe2_symbols
Z[-k1+i2,:]	cIQparameters_log(3,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 -k1+i2 for symbols 1,..., Nprobe2_symbols
Z[-k2+i1,:]	cIQparameters_log(4,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 -k2+i1 for symbols 1,..., Nprobe2_symbols
Z[-k2+i2,:]	cIQparameters_log(5,:)	16x2xNprobe 2_symbols bit	輸出	FFT 輸出, 在窗口 -k2+i2 對於 symbols 1,..., Nprobe2_symbols
$\zeta$	theta	12bit unsigned	輸入	I/Q 補償參數
$\rho$	Rho	12bit	輸入	I/Q 補償參數
Scale_Q	Scale_Q	1bit	輸入	I/Q 補償參數

## 附錄 E

頻率偏移引入、CP 長度、符號數量

```
function [dF] = Frequency_Offset_Introduction(CFO)
```

```
Freq_Th = 41; %round((5e3/50e6)*2*pi*2^16)
```

```
if (abs(CFO)< Freq_Th)
```

```
    dF = sign(CFO)*(TBD RF Interface introduce);
```

```
else
```

```
    dF=0;
```

```
end
```

信號音選擇

```
function [SC] = ProbeII_Tone_Selection(Sigma_2_32)
```

```
SC_MIN(0) = 146;
```

```
SC_MAX(0) = 186;
```

```
SC_MIN(2) = 217;
```

```
SC_MAX(2) = 249;
```

```
SC_DEFAULT(0) = 176;
```

```
SC_DEFAULT(1) = 249;
```

```
NLog_In_Bits=6;
```

```
NLog_Out_Bits=14;
```

```
SC = SC_DEFAULT;
```

```
a=-7871; % round(-1.921625277102556*2^(NLog_Out_Bits-2) );
```

```
b= 8071; % round( 1.970377382221271*2^(NLog_Out_Bits-2) );
```

```
C =32656; %round(0.0375*log2(10)*2^(NLog_Out_Bits-2 +  
NLog_In_Bits));
```

```
%Find Default NSR log2(Sigma_2)
```

```
for i=0:1
```

```
    Scale_16(i) = ceil_log2( Sigma_2_32(SC_DEFAULT(i)) );
```

```
    Frac_16(i) = Sigma_2_32(SC_DEFAULT(i))>> (Scale_16(i)
-NLog_In_Bits );
```

```
    NSR_Default_32(i) = Scale_16(i) << (NLog_Out_Bits - 2 +
NLog_In_Bits)+ ...
```

```
(Real_Mult_16_16(b,Frac_16(i))+a<<(NLog_In_Bits) );
```

```
    NSR_Best_32(i) = NSR_Default_32(i);
```

```
    for k=[SC_MIN(i) : SC_MAX(i)]
```

```
        Scale_k_16 = ceil_log2( Sigma_2_32 (k) );
```

```
        Frac_k_16 = Sigma_2_32 (k)>>(Scale_k_16
-NLog_In_Bits );
```

```
        NSR_32 = Scale_k_16 << (NLog_Out_Bits-2 +
NLog_In_Bits)+ ...
```

```
(Real_Mult_16_16(b,Frac_k_16)+a<<(NLog_In_Bits) );
```

```
        if(NSR_32< Add_Real_32_32(NSR_Default_32 (i),...
- Real_Mult_16_16(C,
Abs_16(Add_Real_16_16(k, -SC_DEFAULT(i)))
```

```

        if(NSR_32 < NSR_Best_32(i))

            SC(i) = k;
            NSR_Best_32(i) = NSR_32;
        end
    end
end
end
end

```

Where Sigma\_32 is a vector of the estimated noise variance of the various FFT tones.

#### CP 和 OFDM 符號數量選擇

```
function [NUM_OF_SYMS, CP_LENGTH] = Probe2_CP_L_Select(CFC
```

```
    CP_Max = 126;
```

```
    CP_Min = 64;
```

```
    L_Max = 40;
```

```
    L_Min = 28;
```

```
    PI = 205887;           %round(pi*2^Freq_bits-1)
```

```
    2P = 411775;          %round(2pi*2^Freq_bits-1)
```

```
    Phase_Th = 41177;     %round(2pi/10*2^Freq_bits-1)
```

```
    dPhase_64= 0;
```

```
    error= 0;
```

```
    Min_Err = Inf;
```

```
    NUM_OF_SYMS = L_Min;
```

```
    CP_LENGTH = CP_Min;
```

```
for CP = CP_Min: CP_Max
    for L= L_Min: L_Max

        dPhase_32 = L*(CP+Nfft)*(CFO);

        while (abs(dPhase_32) > 2P)
            dPhase_32= dPhase_32- sign(dPhase_32)* 2P
        end

        if (abs(dPhase_32 )>PI)
            error_32 = abs(2PI - abs(dPhase_32));
        else
            error_32 =  abs(dPhase_32);
        end

        if( error_32 < Phase_Th )
            Indicator=1;
            if(error_32 <Min_Err)
                Min_Err = error;
                NUM_OF_SYMS = L;
                CP_LENGTH =CP;
            end
        end
    end
end
end
```

**【符號說明】**

資料處理系統 . . .	100
單或多晶片模組 . . .	102
I/O 電路 . . .	104
週邊設備 . . .	106
處理器 . . .	108
記憶體 . . .	110
系統匯流排或別的互連 . . .	112
接收器 . . .	200
射頻(“RF”)處理模組 . . .	202
時域處理模組 . . .	204
頻域處理模組 . . .	206
RF 信號 . . .	208
信號 . . .	209
增益 . . .	210
人為頻率誤差 . . .	213
模數轉換器 . . .	214
失衡補償模組 . . .	218
可變速率內插器 . . .	224
信號 . . .	225
高通濾波器(“HPF”) . . .	228
載波恢復環 . . .	229
接收器控制器 . . .	232
NCO 頻率發生器 . . .	234
數控振盪器(“NCO”)定時發生器 . . .	236
定時信號 . . .	237
前同步碼處理器 . . .	240

載波頻率偏移估計(“CFOE”) . . .	241
CP 移除器 . . .	246
探測 2 軟體處理程式 . . .	250
修正參數 . . .	252
快速傅立葉變換模組 . . .	298
記憶體 . . .	299
接收器 . . .	300
射頻(“RF”)處理模組 . . .	302
時域處理模組 . . .	304
頻域處理模組 . . .	306
RF 信號 . . .	308
信號 . . .	309
增益 . . .	310
人為頻率誤差 . . .	313
模數轉換器 . . .	314
100MHz FIFO(“先進先出”)緩衝器 . . .	316
I/Q 失衡補償模組 . . .	318
基帶模式分層器 . . .	320
自動增益控制器 . . .	322
Farrow 內插器 . . .	324
信號 . . .	325
高通濾波器(“HPF”) . . .	328
時域(“TD”)相位旋轉器 . . .	330
接收器控制器 . . .	332
NCO 頻率發生器 . . .	334
數控振盪器(“NCO”)定時發生器 . . .	336
定時信號 . . .	337
前同步碼控制處理器 . . .	340

載波頻率偏移估計 . . .	341
延遲緩衝器 . . .	342
子電路 . . .	346
探測 2 計算器 . . .	350
探測 2 結果 . . .	352
快速傅立葉變換模組 . . .	398
過程 . . .	600
初始的硬體操作 . . .	602
數控振盪器(“NCO”)相位復位 . . .	604
視窗選擇 . . .	606
殘留頻率誤差估計 . . .	610
數據包 . . .	700

**公告本****發明摘要**

※ 申請案號：96143879

※ 申請日：96.11.20

※ IPC 分類：H04L 27/26 (2006.01)

H03D 3/00 (2006.01)

**【發明名稱】**(中文/英文)

補償資料信號的第一分量和第二分量間失衡的方法及系統 /  
APPARATUS AND METHODS FOR COMPENSATING FOR  
SIGNAL IMBALANCE IN A RECEIVER

**【中文】**

本發明涉及用於補償 I/Q 失衡的裝置、方法和系統，可包括補償資料信號的第一分量和該資料信號的第二分量之間的失衡。通過具有頻率誤差的載波信號可調製該資料信號。可使用至少一參數表徵所述第一分量。該方法可包括接收資料和載波信號；選擇參數值，以便降低負頻率形式的頻率能量；和基於所述值，修改至少一所述分量。

**【英文】**

Apparatus, methods and systems for compensating for an I/Q imbalance may include compensating for an imbalance between a first component of a data signal and a second component of the data signal. The data signal may be modulated by a carrier signal having a frequency error. The first component may be characterized by at least one parameter. The method may include receiving the data and carrier signals; selecting a value for the parameter such that the frequency domain energy at negative frequencies is reduced; and modifying at least one of the components based on the value.

## 申請專利範圍

1. 一種用於補償第一信號和第二信號之間失衡的系統，其特徵在於，包括：

第一模組，用於根據所述第一和第二信號，記錄第一頻域參數、第二頻域參數、第一信號音、第二信號音、以及載波頻率；  
所述第一頻域參數和第二頻域參數相應於所述第一信號音、第二信號音、以及載波頻率中的每一個；和

第二模組，用於根據所述第一和第二頻域參數，計算至少一個時域補償參數；其中，

所述載波頻率是接收器載波頻率；

所述第一和第二信號接收自發送器，所述發送器可使用發送器載波頻率發送所述信號；

由於載波頻率偏移，所述接收器和發送器載波頻率不同；

當所述第一和第二頻域參數相應於所述第一和第二信號音的其中之一時：

所述第一頻域參數對應於離散值頻譜中的第一窗口；

所述第二頻域參數對應於離散值頻譜中的第二窗口；

所述第一窗口鄰近所述第二窗口；

所述第一模組根據所述第一窗口在頻譜中的位置選擇所述第二窗口以及所述第二窗口基於所述偏移，以弧度表示，到  $\pi$  弧度。

2. 如申請專利範圍第 1 項所述的系統，其中，進一步包括：

第三模組，用於根據所述時域補償參數，輸出經補償的信號。

3. 如申請專利範圍第 1 項所述的系統，其中，所述至少一個時域補償參數包括三個時域補償參數。

4. 如申請專利範圍第 1 項所述的系統，其中，當具有殘餘載波頻率偏移，所述第二模組可：

計算殘餘頻率偏移的估計；和

基於所述估計，計算所述時域補償參數。

5. 如申請專利範圍第 1 項所述的系統，其中，當所述第一模組為按時間順序接收到的多個符號中的每一個記錄所述第一和第二頻域參數時，所述第二模組為第一頻域參數計算第一平均值；並為所述第二頻域參數計算第二平均值；所述平均值分別基於與所述符號相應的所述第一和第二頻域參數。
6. 如申請專利範圍第 1 項所述的系統，其中，所述第一模組是硬體模組；所述第二模組是軟體模組。
7. 如申請專利範圍第 2 項所述的系統，其中，所述第三模組是硬體模組。
8. 如申請專利範圍第 1 項所述的系統，其中，所述第一模組是硬體模組；所述第二模組是硬體模組。

**【代表圖】**

【本案指定代表圖】：圖 2。

【本代表圖之符號簡單說明】：

接收器	200
射頻(“RF”)處理模組	202
時域處理模組	204
頻域處理模組	206
RF 信號	208
信號	209
增益	210
人為頻率誤差	213
模數轉換器	214
失衡補償模組	218
可變速率內插器	224
信號	225
高通濾波器(“HPF”)	228
載波恢復環	229
接收器控制器	232
NCO 頻率發生器	234
數控振盪器(“NCO”)定時發生器	236
定時信號	237
前同步碼處理器	240
載波頻率偏移估計(“CFOE”)	241
CP 移除器	246
探測 2 軟體處理程式	250
修正參數	252
快速傅立葉變換模組	298
記憶體	299

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：