



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0026640

(51)⁷ G10L 19/008; H04S 3/00

(13) B

(21) 1-2016-01174

(22) 26/09/2014

(86) PCT/US2014/057611 26/09/2014

(87) WO 2015/048387 A1 02/04/2015

(30) 61/883,890 27/09/2013 US

(45) 25/12/2020 393

(43) 27/06/2016 339A

(73) DOLBY LABORATORIES LICENSING CORPORATION (US)

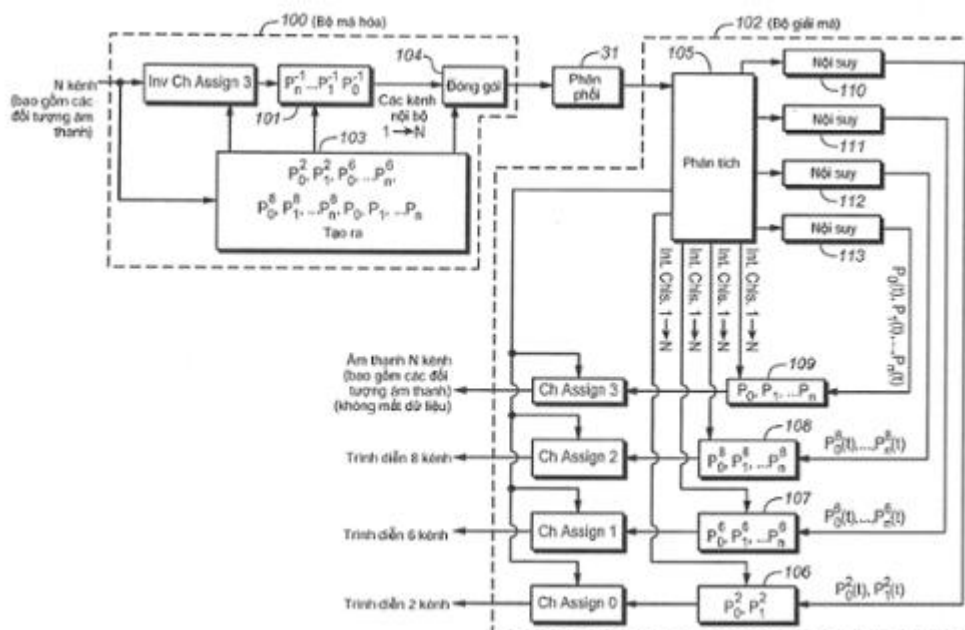
1275 Market Street, San Francisco, California 94103, United States of America.

(72) LAW, Malcolm J. (GB); MELKOTE, Vinay (IN); WILSON, Rhonda (GB); PLAIN, Simon (CA); JASPAR, Andy (US).

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP MÃ HÓA CHƯƠNG TRÌNH ÂM THANH N KÊNH VÀ PHƯƠNG PHÁP PHỤC HỒI M KÊNH CỦA CHƯƠNG TRÌNH ÂM THANH N KÊNH, BỘ MÃ HÓA VÀ BỘ GIẢI MÃ ÂM THANH

(57) Sáng chế đề cập đến phương pháp mã hóa chương trình âm thanh N kênh và phục hồi M kênh của chương trình âm thanh N kênh, bộ mã hóa và bộ giải mã âm thanh. Các phương pháp mà sử dụng các ma trận nguyên thủy được nội suy để giải mã âm thanh được mã hóa để phục hồi (không mất dữ liệu) nội dung của chương trình âm thanh đa kênh và/hoặc để phục hồi ít nhất một dạng trộn giảm của nội dung đó, và phương pháp mã hóa để tạo ra âm thanh được mã hóa như vậy. Theo một số phương án, bộ giải mã thực hiện phép nội suy đối với tập hợp các ma trận hạt giống nguyên thủy để xác định các ma trận được nội suy dùng để kết xuất các kênh của chương trình. Các khía cạnh khác là hệ thống hoặc thiết bị được tạo cấu hình để thực hiện phương pháp theo phương án bất kỳ.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc xử lý tín hiệu âm thanh, và cụ thể hơn đến việc kết xuất chương trình âm thanh đa kênh (ví dụ, các dòng bit biểu thị các chương trình âm thanh dựa trên đối tượng bao gồm ít nhất một kênh đối tượng âm thanh và ít nhất một kênh loa) nhờ sử dụng các ma trận được nội suy, và đến việc mã hóa và giải mã các chương trình này. Theo một số phương án, bộ giải mã thực hiện phép nội suy trên tập hợp ma trận hạt giống nguyên thủy để xác định các ma trận được nội suy dùng để kết xuất các kênh của chương trình. Một số phương án tạo ra, giải mã, và/hoặc kết xuất dữ liệu âm thanh theo định dạng được gọi là Dolby TrueHD.

Tình trạng kỹ thuật của sáng chế

Dolby và Dolby TrueHD là các nhãn hiệu của hãng Dolby Laboratories Licensing Corporation.

Tính phức tạp, và chi phí tài chính và tính toán, của việc kết xuất các chương trình âm thanh tăng theo số lượng kênh cần được kết xuất. Trong quá trình kết xuất và phát lại các chương trình âm thanh dựa trên đối tượng, nội dung âm thanh có nhiều kênh (ví dụ, kênh đối tượng và kênh loa) thường lớn hơn nhiều (ví dụ, ở mức tăng khuếch đại) so với số lượng kênh có trong quá trình kết xuất và phát lại các chương trình dựa trên kênh loa thông thường. Cũng thông thường, hệ thống loa dùng để phát lại bao gồm số lượng loa lớn hơn nhiều so với số lượng loa dùng để phát lại các chương trình dựa trên kênh loa thông thường.

Mặc dù các phương án của sáng chế là hữu dụng để kết xuất các kênh của chương trình âm thanh đa kênh bất kỳ, nhưng nhiều phương án của sáng chế đặc biệt hữu dụng để kết xuất các kênh của các chương trình âm thanh dựa trên đối tượng có số lượng kênh lớn.

Đã biết sử dụng các hệ thống phát lại (ví dụ, ở các rạp chiếu phim) để kết xuất các chương trình âm thanh dựa trên đối tượng. Các chương trình âm thanh dựa trên đối tượng có thể biểu thị nhiều đối tượng âm thanh khác nhau tương ứng với các hình ảnh trên màn hình, cuộc đối thoại, tạp âm, và các hiệu ứng âm thanh mà bắt nguồn từ

nhiều vị trí khác nhau trên (hoặc tương ứng với) màn hình, cũng như hiệu ứng âm nhạc đệm và hiệu ứng môi trường xung quanh (mà có thể được biểu thị bởi các kênh loa của chương trình) để tạo ra trải nghiệm thính giác đầy đủ được dự định. Việc phát lại chính xác các chương trình như vậy đòi hỏi âm thanh được tái tạo theo cách tương đương gần giống nhất có thể với những gì được người sáng tạo nội dung dự định về kích thước đối tượng, vị trí, cường độ, chuyển động và độ sâu của âm thanh.

Trong quá trình tạo ra các chương trình âm thanh dựa trên đối tượng, các loa cần dùng để kết xuất thường được giả định đặt ở các vị trí tùy ý trong môi trường phát lại; không nhất thiết phải đặt theo sự bố trí định trước trên mặt phẳng ngang (danh nghĩa) hoặc theo sự bố trí định trước khác bất kỳ được biết vào thời điểm tạo ra chương trình. Thông thường, siêu dữ liệu bao gồm trong chương trình biểu thị các thông số kết xuất để kết xuất ít nhất một đối tượng của chương trình ở vị trí không gian rõ ràng hoặc dọc theo quỹ đạo (theo thể tích ba chiều), ví dụ, sử dụng mảng loa ba chiều. Ví dụ, kênh đối tượng của chương trình có thể có siêu dữ liệu tương ứng biểu thị quỹ đạo ba chiều của các vị trí không gian rõ ràng mà ở đó đối tượng (được biểu thị bởi kênh đối tượng) sẽ được kết xuất. Quỹ đạo có thể bao gồm dãy các vị trí "sàn" (trên mặt phẳng của tập hợp con các loa mà giả định được đặt trên sàn, hoặc trên mặt phẳng ngang khác, của môi trường phát lại), và dãy các vị trí "trên sàn" (mỗi vị trí được xác định bằng cách điều khiển tập hợp con các loa mà được giả định sẽ đặt trên ít nhất một mặt phẳng ngang khác của môi trường phát lại).

Các chương trình âm thanh dựa trên đối tượng thể hiện sự cải thiện đáng kể ở nhiều khía cạnh so với các chương trình âm thanh dựa trên kênh loa thông thường, vì âm thanh dựa trên kênh loa bị giới hạn hơn ở việc phát lại không gian các đối tượng âm thanh cụ thể so với âm thanh dựa trên kênh đối tượng. Các chương trình âm thanh dựa trên kênh loa chỉ bao gồm các kênh loa (không bao gồm kênh đối tượng), và mỗi kênh loa thường xác định tín hiệu nạp vào loa cho loa riêng lẻ, cụ thể trong môi trường nghe.

Các phương pháp và hệ thống khác nhau để tạo ra và kết xuất các chương trình âm thanh dựa trên đối tượng đã được đề xuất. Trong quá trình tạo ra chương trình âm thanh dựa trên đối tượng, thông thường giả định rằng số lượng tùy ý các loa sẽ được sử dụng để phát lại chương trình, và rằng các loa cần sử dụng để phát lại này sẽ nằm ở các vị trí tùy ý trong môi trường phát lại; chứ không nhất thiết phải ở mặt phẳng

ngang (danh nghĩa) hoặc theo sự sắp đặt định trước khác bất kỳ được biết tại thời điểm tạo ra chương trình. Thông thường, siêu dữ liệu về đối tượng được đưa vào chương trình biểu thị các thông số kết xuất để kết xuất ít nhất một đối tượng của chương trình ở vị trí không gian rõ ràng hoặc dọc theo một quỹ đạo (theo thể tích ba chiều), ví dụ, sử dụng mảng các loa ba chiều. Ví dụ, kênh đối tượng của chương trình có thể có siêu dữ liệu tương ứng biểu thị quỹ đạo ba chiều của các vị trí không gian rõ ràng mà đối tượng (được biểu thị bởi kênh đối tượng) sẽ được kết xuất ở đó. Quỹ đạo có thể bao gồm dãy các vị trí “sàn” (trên mặt phẳng của tập hợp con các loa mà được giả định là được đặt trên sàn, hoặc trên một mặt phẳng ngang khác, của môi trường phát lại), và dãy các vị trí “trên sàn” (mỗi vị trí được xác định bằng cách điều khiển tập hợp con của các loa mà được giả định là được đặt ở ít nhất một mặt phẳng nằm ngang khác của môi trường phát lại). Các ví dụ, về kết xuất các chương trình âm thanh dựa trên đối tượng được mô tả, ví dụ, trong đơn quốc tế PCT số PCT/US2001/028783, được công bố theo công bố quốc tế số WO 2011/119401 A2 ngày 29.09.2011, và được chuyển nhượng cho người nhận chuyển nhượng của đơn này.

Chương trình âm thanh dựa trên đối tượng có thể bao gồm các kênh “đệm”. Kênh đệm có thể là kênh đối tượng biểu thị đối tượng có vị trí không thay đổi theo khoảng thời gian thích hợp (và vì vậy thường được kết xuất bằng cách sử dụng tập hợp loa của hệ thống phát lại có các vị trí loa tĩnh), hoặc nó có thể là kênh loa (cần được kết xuất bởi loa cụ thể của hệ thống phát lại). Các kênh đệm không có siêu dữ liệu về vị trí thay đổi theo thời gian tương ứng (cho dù chúng có thể được coi là có siêu dữ liệu về vị trí không thay đổi theo thời gian). Chúng có thể biểu thị các phần tử âm thanh mà được phân tán trong không gian, ví dụ, âm thanh biểu thị môi trường xung quanh.

Việc phát lại chương trình âm thanh dựa trên đối tượng nhờ lắp đặt loa truyền thống (ví dụ hệ thống phát lại 7.1) đạt được bằng cách kết xuất các kênh của chương trình (bao gồm các kênh đối tượng) thành tập hợp tín hiệu nạp cho loa. Theo các phương án thông thường của sáng chế, quy trình kết xuất các kênh đối tượng (đôi khi được gọi trong bản mô tả này là các đối tượng) và các kênh khác của chương trình âm thanh dựa trên đối tượng (hoặc các kênh của chương trình âm thanh thuộc loại khác) bao gồm chủ yếu (hoặc duy nhất) sự chuyển đổi siêu dữ liệu không gian (đối với các

kênh cần được kết xuất) ở mỗi thời điểm thời gian thành ma trận độ lợi tương ứng (được gọi trong bản mô tả này là “ma trận kết xuất”) mà biểu diễn mỗi kênh (ví dụ, các kênh đối tượng và các kênh loa) nên đóng góp bao nhiêu vào hỗn hợp gồm nội dung âm thanh (theo thời điểm) được biểu thị bởi tín hiệu nạp vào loa cho loa cụ thể (ví dụ, trọng lượng tương đối của mỗi kênh của chương trình trong hỗn hợp được biểu thị bởi tín hiệu nạp vào loa).

“Kênh đối tượng” của chương trình âm thanh dựa trên đối tượng biểu thị dãy các mẫu biểu thị đối tượng âm thanh, và chương trình thường bao gồm dãy các giá trị siêu dữ liệu về vị trí không gian biểu thị vị trí đối tượng hoặc quỹ đạo đối với mỗi kênh đối tượng. Theo các phương án thông thường của sáng chế, các dãy của các giá trị siêu dữ liệu về vị trí tương ứng với các kênh đối tượng của chương trình được sử dụng để xác định ma trận $M \times N$ $A(t)$ biểu thị đặc tả độ lợi thay đổi theo thời gian đối với chương trình.

Việc kết xuất “N” kênh (ví dụ, các kênh đối tượng, hoặc các kênh đối tượng và các kênh loa) của chương trình âm thanh cho “M” loa (các tín hiệu nạp vào loa) ở thời điểm “t” của chương trình có thể được biểu diễn bằng cách nhân vector $\mathbf{x}(t)$ có độ dài “N”, bao gồm mẫu âm thanh ở thời điểm “t” từ mỗi kênh, với ma trận $M \times N$ $A(t)$ được xác định từ siêu dữ liệu về vị trí liên quan (và tùy ý từ siêu dữ liệu khác tương ứng với nội dung âm thanh cần được kết xuất, ví dụ, các độ lợi đối tượng) ở thời điểm “t”. Các giá trị thu được (ví dụ, các độ lợi hoặc mức độ) của các tín hiệu nạp vào loa ở thời điểm t có thể được biểu diễn dưới dạng vector $\mathbf{y}(t)$, như trong phương trình sau đây (1):

$$\begin{bmatrix} y_0(t) \\ y_1(t) \\ \vdots \\ y_{M-1}(t) \end{bmatrix} = \begin{bmatrix} a_{00}(t) & a_{01}(t) & a_{02}(t) & \ddots & a_{0,N-1}(t) \\ a_{10}(t) & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ a_{M-1,0}(t) & \ddots & \ddots & \ddots & a_{M-1,N-1}(t) \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ \vdots \\ x_{N-1}(t) \end{bmatrix}$$

$$\mathbf{y}(t) \quad \mathbf{A}(t) \quad \mathbf{x}(t) \quad . \quad (1)$$

Mặc dù phương trình (1) mô tả việc kết xuất N kênh của chương trình âm thanh (ví dụ, chương trình âm thanh dựa trên đối tượng, hoặc phiên bản được mã hóa của chương trình âm thanh dựa trên đối tượng) thành M kênh đầu ra (ví dụ, M tín hiệu nạp vào loa), nhưng nó cũng biểu diễn tập hợp tình huống chung trong đó tập

hợp N mẫu âm thanh được biến đổi thành tập hợp M giá trị (ví dụ, M mẫu) bằng các phép toán tuyến tính. Ví dụ, $A(t)$ có thể là ma trận tĩnh, “A”, có các hệ số không thay đổi theo các giá trị thời gian khác nhau “t”. Một ví dụ, khác, $A(t)$ (mà có thể là ma trận tĩnh, A) có thể biểu diễn tín hiệu giải mã thông thường của tập hợp các kênh loa $x(t)$ với tập hợp các kênh loa nhỏ hơn $y(t)$ (hoặc $x(t)$ có thể là tập hợp của các kênh âm thanh mô tả cảnh không gian theo định dạng Ambisonics), và sự chuyển đổi thành các tín hiệu nạp vào loa $y(t)$ có thể được quy định là phép nhân với ma trận trộn giảm A. Thậm chí trong ứng dụng sử dụng ma trận trộn giảm tĩnh danh định, sự biến đổi tuyến tính thực (nhân ma trận) được áp dụng có thể là động để bảo đảm sự bảo vệ kẹp của quá trình trộn giảm (ví dụ, ma trận biến đổi tĩnh A có thể được chuyển đổi thành ma trận biến đổi thay đổi theo thời gian $A(t)$, để bảo đảm sự bảo vệ kẹp).

Hệ thống kết xuất chương trình âm thanh (ví dụ, bộ giải mã thực thi hệ thống này) có thể nhận siêu dữ liệu để xác định chỉ các ma trận kết xuất $A(t)$ (hoặc nó có thể nhận chính các ma trận) không liên tục và không phải ở mọi thời điểm “t” trong suốt chương trình. Ví dụ, việc này có thể là do bất kỳ trong số các lý do, ví dụ, độ phân giải theo thời gian thấp của hệ thống mà thực sự kết xuất siêu dữ liệu hoặc nhu cầu giới hạn tốc độ truyền bit của chương trình. Các tác giả sáng chế đã nhận ra rằng có thể là mong muốn đối với hệ thống kết xuất nếu nội suy giữa các ma trận kết xuất $A(t_1)$ và $A(t_2)$, lần lượt ở các thời điểm “t1” và “t2” trong suốt chương trình, để thu được ma trận kết xuất $A(t_3)$ cho thời điểm trung gian “t3.” Phép nội suy bảo đảm rằng vị trí được nhận biết của các đối tượng trong tín hiệu nạp vào loa được kết xuất thay đổi đều theo thời gian, và có thể loại bỏ nhiều lạ không mong muốn như tiếng mở khóa kéo bắt nguồn từ việc cập nhật ma trận không liên tục (không đổi theo từng phần). Phép nội suy có thể là tuyến tính (hoặc không tuyến tính), và thường phải bảo đảm một đường liên tục theo thời gian từ $A(t_1)$ đến $A(t_2)$.

Dolby TrueHD là định dạng mã hóa-giải mã âm thanh thông thường mà hỗ trợ cuộc truyền không tổn hao và có thể thay đổi tỷ lệ tín hiệu âm thanh. Âm thanh nguồn được mã hóa thành thứ tầng của các dòng phụ của các kênh, và tập hợp con được chọn của các dòng phụ (chứ không phải tất cả các dòng phụ) có thể được phục hồi từ dòng bit và được giải mã, để thu được biểu diễn kích thước thấp hơn (trộn giảm) của cảnh không gian. Khi tất cả các dòng phụ được giải mã, âm thanh tổng hợp giống với âm thanh nguồn (việc mã hóa, tiếp đó là giải mã, là không mất dữ liệu).

Trong phiên bản TrueHD có sẵn trên thị trường, âm thanh nguồn thường là âm thanh trộn kênh 7.1 mà được mã hóa thành dãy ba dòng phụ, bao gồm dòng phụ thứ nhất mà có thể được giải mã để xác định hai tín hiệu trộn giảm kênh của âm thanh gốc kênh 7.1. Hai dòng phụ thứ nhất có thể được giải mã để xác định tín hiệu trộn giảm kênh 5.1 của âm thanh gốc. Tất cả ba dòng phụ có thể được giải mã để xác định âm thanh gốc kênh 7.1. Chi tiết kỹ thuật của công nghệ Dolby TrueHD, và công nghệ đóng gói không mất dữ liệu chuẩn (Meridian Lossless Packing - MLP) mà nó dựa vào, được biết rộng rãi. Các khía cạnh của công nghệ TrueHD và MLP được mô tả trong bằng sáng chế Mỹ số 6,611,212 cấp ngày 26.08.2003, và được chuyển nhượng cho Dolby Laboratories Licensing Corp., và tài liệu của Gerzon, et al., có tựa đề “The MLP Lossless Compression System for PCM Audio,” J. AES, Vol. 52, No. 3, pp. 243-260 (03.2004).

TrueHD hỗ trợ đặc tả của các ma trận trộn giảm. Trong quá trình sử dụng thông thường, bộ tạo nội dung của chương trình âm thanh kênh 7.1 định rõ ma trận tĩnh để trộn giảm chương trình kênh 7.1 về ma trận trộn giảm kênh 5.1, và ma trận tĩnh khác để trộn giảm ma trận trộn giảm kênh 5.1 về ma trận trộn giảm 2 kênh. Mỗi ma trận trộn giảm tĩnh có thể được biến đổi thành dãy các ma trận trộn giảm (mỗi ma trận trong dãy trộn giảm một khoảng thời gian khác nhau trong chương trình) để đạt được sự bảo vệ kịp. Tuy nhiên, mỗi ma trận trong dãy được truyền (hoặc siêu dữ liệu xác định mỗi ma trận trong dãy được truyền) cho bộ giải mã, và bộ giải mã không thực hiện phép nội suy trên bất kỳ ma trận trộn giảm được định rõ trước đó để xác định ma trận tiếp theo trong dãy các ma trận trộn giảm cho chương trình.

Fig.1 là sơ đồ các phần tử của hệ thống TrueHD thông thường, trong đó bộ mã hóa (30) và bộ giải mã (32) được tạo cấu hình để thực hiện các phép toán lập ma trận trên các mẫu âm thanh. Trong hệ thống trên Fig.1, bộ mã hóa 30 được tạo cấu hình để mã hóa chương trình âm thanh 8 kênh (ví dụ, tập hợp tín hiệu nạp vào loa 7.1 truyền thống) dưới dạng dòng bit được mã hóa bao gồm hai dòng phụ, và bộ giải mã 32 được tạo cấu hình để giải mã dòng bit được mã hóa để kết xuất chương trình 8 kênh gốc (theo cách không mất dữ liệu) hoặc dạng trộn giảm 2 kênh của chương trình 8 kênh gốc. Bộ mã hóa 30 được ghép nối và được tạo cấu hình để tạo ra dòng bit mã hóa và để xác nhận dòng bit mã hóa cho hệ thống phân phối 31.

Hệ thống phân phối 31 được ghép nối và được tạo cấu hình để phân phối (ví dụ, bằng cách lưu trữ và/hoặc truyền) dòng bit mã hóa cho bộ giải mã 32. Theo một số phương án, hệ thống 31 thực hiện việc phân phối (ví dụ, truyền) chương trình âm thanh đa kênh được mã hóa trên hệ thống phát rộng hoặc mạng (ví dụ, internet) đến bộ giải mã 32. Theo một số phương án, hệ thống 31 lưu trữ chương trình âm thanh đa kênh được mã hóa trong phương tiện lưu trữ (ví dụ, đĩa hoặc bộ đĩa), và bộ giải mã 32 được tạo cấu hình để đọc chương trình từ phương tiện lưu trữ này.

Khối được đánh dấu là “InvChAssign1” trong bộ mã hóa 30 được tạo cấu hình để thực hiện hoán vị kênh (tương đương phép nhân với ma trận hoán vị) trên các kênh của chương trình đầu vào. Sau đó, các kênh được hoán vị trải qua quá trình mã hóa ở bước 33, bước này xuất ra tám kênh tín hiệu mã hóa. Các kênh tín hiệu mã hóa này có thể (nhưng không cần thiết) tương ứng với các kênh loa phát lại. Các kênh tín hiệu mã hóa đôi khi được gọi là các kênh “nội bộ” vì bộ giải mã (và/hoặc hệ thống kết xuất) thường giải mã và kết xuất nội dung của các kênh tín hiệu mã hóa để phục hồi âm thanh đầu vào, sao cho các kênh tín hiệu mã hóa là các kênh “nội bộ” đối với hệ thống mã hóa/giải mã. Quá trình mã hóa được thực hiện ở bước 33 tương đương với phép nhân mỗi tập hợp mẫu của các kênh hoán vị với ma trận mã hóa (được thực hiện như một tầng $n+1$ phép nhân ma trận, được nhận dạng là $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$, sẽ được mô tả chi tiết hơn dưới đây).

Hệ thống phụ xác định ma trận 34 được tạo cấu hình để tạo ra dữ liệu biểu thị các hệ số của hai tập hợp ma trận đầu ra (một tập hợp tương ứng với mỗi trong số hai dòng phụ của các kênh mã hóa). Một tập hợp các ma trận đầu ra bao gồm hai ma trận, P_0^2, P_1^2 , mỗi trong số này là ma trận nguyên thủy (được định nghĩa dưới đây) có kích thước 2×2 , và được dùng để kết xuất dòng phụ thứ nhất (dòng phụ trộn giảm) bao gồm hai trong số các kênh âm thanh mã hóa của dòng bit mã hóa (để kết xuất dạng trộn giảm hai kênh của âm thanh đầu vào tám kênh). Tập hợp ma trận đầu ra còn lại bao gồm các ma trận kết xuất, $P_0, P_1, \dots, P_n$, mỗi trong số đó là ma trận nguyên thủy, và được dùng để kết xuất dòng phụ thứ hai bao gồm tất cả tám kênh âm thanh mã hóa của dòng bit mã hóa (để phục hồi không mất dữ liệu cho chương trình âm thanh đầu vào tám kênh). Tầng các ma trận, P_0^2, P_1^2 , cùng với các ma trận $P_0^{-1}, P_1^{-1}, \dots, P_n^{-1}$, áp dụng cho âm thanh ở bộ mã hóa, bằng đặc tả của ma trận trộn giảm mà biến đổi 8

kênh âm thanh đầu vào thành dạng trộn giảm 2 kênh, và tăng các ma trận, $P_0, P_1, \dots, P_n$, kết xuất ngược 8 kênh mã hóa của dòng bit mã hóa vào 8 kênh đầu vào gốc.

Các hệ số (của mỗi ma trận) mà được xuất ra từ hệ thống phụ 34 cho hệ thống phụ đóng gói 35 là siêu dữ liệu biểu thị độ lợi tương đối hoặc tuyệt đối của mỗi kênh sẽ được bao gồm trong dạng trộn kênh tương ứng của chương trình. Các hệ số của mỗi ma trận kết xuất (trong một thời điểm trong suốt chương trình) biểu diễn mỗi kênh nên đóng góp bao nhiêu vào dạng trộn nội dung âm thanh (ở thời điểm tương ứng của dạng trộn được kết xuất) được biểu thị bởi tín hiệu nạp vào loa cho loa của hệ thống phát lại riêng.

Tám kênh âm thanh mã hóa (xuất ra từ bước mã hóa 33), các hệ số ma trận đầu ra (được tạo ra bởi hệ thống phụ 34), và thông thường dữ liệu bổ sung cũng được xác nhận với hệ thống phụ đóng gói 35, hệ thống này tập hợp chúng thành dòng bit mã hóa mà sau đó được xác thực với hệ thống phân phối 31.

Dòng bit được mã hóa bao gồm dữ liệu biểu thị tám kênh âm thanh được mã hóa, hai tập hợp ma trận đầu ra (một tập hợp tương ứng với mỗi trong số hai dòng phụ của các kênh được mã hóa), và thường bao gồm cả dữ liệu bổ sung (ví dụ, siêu dữ liệu về nội dung âm thanh).

Hệ thống phụ phân tích 36 của bộ giải mã 32 được tạo cấu hình để chấp nhận (đọc hoặc nhận) dòng bit được mã hóa từ hệ thống phân phối 31 và để phân tích dòng bit được mã hóa này. Hệ thống phụ 36 có thể hoạt động để xác nhận các dòng phụ của dòng bit được mã hóa, bao gồm dòng phụ “thứ nhất” bao gồm chỉ hai trong số các kênh được mã hóa của dòng bit được mã hóa, và các ma trận đầu ra (P_0^2, P_1^2) tương ứng với dòng phụ thứ nhất, cho bước nhân ma trận 38 (để xử lý dòng phụ mà dẫn đến sự biểu diễn trộn giảm 2 kênh nội dung của chương trình đầu vào 8 kênh gốc). Hệ thống phụ 36 cũng có thể hoạt động để xác nhận các dòng phụ của dòng bit được mã hóa (dòng phụ “thứ hai” bao gồm tất cả tám kênh được mã hóa của dòng bit được mã hóa) và các ma trận đầu ra tương ứng ($P_0, P_1, \dots, P_n$) cho bước nhân ma trận 37 để xử lý dòng phụ mà dẫn đến việc kết xuất chương trình gốc 8 kênh theo cách không mất dữ liệu.

Cụ thể hơn, bước 38 nhân hai mẫu âm thanh của hai kênh của dòng phụ thứ nhất với dãy các ma trận P_0^2, P_1^2 , và mỗi tập hợp kết quả của hai mẫu biến đổi tuyến

tính trải qua quá trình hoán vị kênh (tương đương việc nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign0” để tạo ra mỗi cặp mẫu của dạng trộn giảm 2 kênh yêu cầu của 8 kênh âm thanh gốc. Dãy phép toán ma trận được thực hiện trong bộ mã hóa 30 và bộ giải mã 32 tương đương với việc áp dụng đặc tả ma trận trộn giảm mà biến đổi 8 kênh âm thanh đầu vào thành dạng trộn giảm 2 kênh.

Bước 37 nhân mỗi vector gồm tám mẫu âm thanh (một mẫu lấy từ mỗi trong số tập hợp đầy đủ tám kênh của dòng bit được mã hóa) với dãy các ma trận $P_0, P_1, \dots, P_n$, và mỗi tập hợp kết quả gồm tám mẫu biến đổi tuyến tính trải qua quá trình hoán vị kênh (tương đương việc nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign1” để tạo ra mỗi tập hợp gồm tám mẫu của chương trình tám kênh gốc phục hồi được không mất dữ liệu. Để cho âm thanh 8 kênh đầu ra là giống chính xác như âm thanh 8 kênh đầu vào (để đạt được đặc điểm “không mất dữ liệu” của hệ thống), các phép toán lập ma trận được thực hiện trong bộ mã hóa 30 cần chính xác (bao gồm các hiệu ứng lượng tử hóa) ngược với các phép toán lập ma trận được thực hiện trong bộ giải mã 32 trên dòng phụ (thứ hai) không mất dữ liệu của dòng bit được mã hóa (tức là, nhân với tăng của ma trận $P_0, P_1, \dots, P_n$). Vì vậy, trên Fig.1, các phép toán lập ma trận ở bước 33 của bộ mã hóa 30 được nhận dạng là tăng của ma trận nghịch đảo của các ma trận $P_0, P_1, \dots, P_n$, trong dãy đối diện được áp dụng ở bước 37 của bộ giải mã 32, tức là: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

Bộ giải mã 32 áp dụng phép nghịch đảo hoán vị kênh được áp dụng bởi bộ mã hóa 30 (tức là, ma trận hoán vị được biểu diễn bởi phần tử “ChAssign1” của bộ giải mã 32 nghịch đảo với ma trận được biểu diễn bởi phần tử “InvChAssign1” của bộ mã hóa 30).

Căn cứ vào đặc tả ma trận trộn giảm (ví dụ, đặc tả ma trận tĩnh A có kích thước 2×8), mục tiêu của việc thực hiện mã hóa TrueHD thông thường của bộ mã hóa 30 là thiết kế các ma trận đầu ra (ví dụ, $P_0, P_1, \dots, P_n$ và P_0^2, P_1^2 trên Fig.1), và các ma trận đầu vào ($P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$) và các phần gán kênh đầu ra (và đầu vào) sao cho:

1. dòng bit được mã hóa là phân cấp (ví dụ, trong ví dụ, hai kênh mã hóa thứ nhất đủ để suy ra biểu diễn trộn giảm 2 kênh, và tập hợp đầy đủ tám kênh được mã hóa đủ để phục hồi chương trình 8 kênh gốc); và

2. các ma trận cho dòng cao nhất ($P_0, P_1, \dots, P_n$ trong ví dụ) có thể nghịch đảo được sao cho âm thanh đầu vào có thể phục hồi được chính xác bởi bộ giải mã.

Các hệ thống tính toán thông thường hoạt động với độ chính xác hữu hạn và việc đảo ma trận khả nghịch tùy ý một cách chính xác có thể cần độ chính xác rất lớn. TrueHD khắc phục vấn đề này bằng cách ép buộc các ma trận đầu ra và các ma trận đầu vào (tức là, $P_0, P_1, \dots, P_n$ và $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$) là các ma trận vuông thuộc loại được biết đến là “các ma trận nguyên thủy”.

Ma trận nguyên thủy P có kích thước $N \times N$ có dạng:

$$P = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \ddots \\ \alpha_0 & \alpha_1 & \alpha_2 & \ddots & \alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Ma trận nguyên thủy luôn là ma trận hình vuông. Ma trận nguyên thủy kích thước $N \times N$ giống với ma trận đồng nhất kích thước $N \times N$ ngoại trừ một hàng (tầm thường) (tức là, hàng bao gồm các phần tử $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ trong ví dụ). Trong tất cả các hàng khác, các phần tử ngoài đường chéo bằng 0 và phần tử chung đường chéo có giá trị tuyệt đối 1 (tức là, hoặc +1 hoặc -1). Để đơn giản hóa ngôn ngữ trong bản mô tả này, các hình vẽ và phần mô tả sẽ luôn giả định rằng ma trận nguyên thủy có các phần tử đường chéo bằng +1 với khả năng ngoại trừ phần tử đường chéo ở hàng không quan trọng. Tuy nhiên, cần lưu ý rằng việc này không làm mất tính tổng quát, và các ý tưởng được trình bày trong bản mô tả này gắn với nhóm chung của các ma trận nguyên thủy trong đó các phần tử chéo có thể bằng +1 hoặc -1.

Khi ma trận nguyên thủy, P , hoạt động trên (ví dụ, nhân) vector $x(t)$, kết quả là tích $Px(t)$, tích này là một vector kích thước N khác mà giống chính xác như vector $x(t)$ trong tất cả các phần tử ngoại trừ một. Do đó, mỗi ma trận nguyên thủy có thể được kết hợp với một kênh duy nhất mà ma trận đó vận dụng được (hoặc ma trận hoạt động trên đó).

Sáng chế sẽ sử dụng thuật ngữ “ma trận nguyên thủy dạng đơn vị” trong bản mô tả này để chỉ ma trận nguyên thủy trong đó phần tử dùng chung đường chéo (của hàng không tầm thường của ma trận nguyên thủy) có giá trị tuyệt đối 1 (ví dụ, +1

hoặc -1). Vì vậy, đường chéo của ma trận nguyên thủy dạng đơn vị bao gồm tất cả các giá trị dương, +1, hoặc tất cả các giá trị âm, -1, hoặc một số giá trị dương và một số giá trị âm. Ma trận nguyên thủy chỉ thay đổi một kênh của tập hợp (vector) gồm các mẫu của các kênh chương trình âm thanh, và ma trận nguyên thủy dạng đơn vị cũng là ma trận khả nghịch theo cách không mất dữ liệu do các giá trị đơn vị trên đường chéo. Hơn nữa, để đơn giản hóa phần mô tả ở đây, sáng chế sẽ sử dụng thuật ngữ ma trận nguyên thủy dạng đơn vị để chỉ ma trận nguyên thủy mà hàng không tầm thường của nó có phần tử chéo +1. Tuy nhiên, mọi sự liên quan đến ma trận nguyên thủy dạng đơn vị trong bản mô tả này, bao gồm trong phần yêu cầu bảo hộ, được dự định để bao gồm trường hợp chung hơn trong đó ma trận nguyên thủy dạng đơn vị có thể có hàng không tầm thường mà phần tử chung của nó có đường chéo là +1 hoặc -1.

Nếu $\alpha_2=1$ (dẫn đến ma trận nguyên thủy dạng đơn vị có đường chéo bao gồm các giá trị dương) ở ví dụ, trên trong ma trận nguyên thủy, P, thì thấy rằng nghịch đảo của P chính xác là:

$$P^{-1} = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \ddots \\ -\alpha_0 & -\alpha_1 & 1 & \ddots & -\alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Nói chung đúng là nghịch đảo của ma trận nguyên thủy dạng đơn vị được xác định một cách đơn giản bằng cách đảo (nhân với -1) mỗi trong số các hệ số α không tầm thường của nó mà không nằm dọc theo đường chéo.

Nếu các ma trận $P_0, P_1, \dots, P_n$ được sử dụng trong bộ giải mã 32 trên Fig.1 là các ma trận nguyên thủy dạng đơn vị (có các đường chéo đơn vị), thì dãy phép toán lập ma trận $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$ trong bộ mã hóa 30 và $P_0, P_1, \dots, P_n$ trong bộ giải mã 32 có thể được thực hiện bởi các mạch có độ chính xác hữu hạn thuộc loại được thể hiện trên Fig.2A và Fig.2B. Fig.2A là hệ mạch thông thường của bộ mã hóa thực hiện việc lập ma trận không mất dữ liệu thông qua các ma trận nguyên thủy được thực hiện bởi thuật toán có độ chính xác hữu hạn. Fig.2B là hệ mạch thông thường của bộ giải mã để thực hiện việc lập ma trận không mất dữ liệu thông qua các ma trận nguyên thủy được thực hiện bởi thuật toán có độ chính xác hữu hạn. Chi tiết về các dạng thực hiện điển

hình của hệ mạch trên Fig.2A và Fig.2B (và các biến đổi trên đó) được mô tả trong sáng chế Mỹ nêu trên số 6,611,212, cấp ngày 26.08.2003.

Trên Fig.2A (thể hiện hệ mạch để mã hóa chương trình âm thanh bốn kênh bao gồm các kênh S1, S2, S3, và S4), ma trận nguyên thủy thứ nhất P_0^{-1} (có một hàng gồm bốn hệ số α khác 0) hoạt động trên mỗi mẫu của kênh S1 (để tạo ra kênh mã hóa S1') bằng cách trộn mẫu liên quan của kênh S1 với các mẫu tương ứng (xảy ra cùng thời gian, t) của các kênh S2, S3, và S4. Ma trận nguyên thủy thứ hai P_1^{-1} (cũng có một hàng gồm bốn hệ số α khác 0) hoạt động trên mỗi mẫu của kênh S2 (để tạo ra mẫu tương ứng của kênh mã hóa S2') bằng cách trộn mẫu liên quan của kênh S2 với các mẫu tương ứng của các kênh S1', S3, và S4. Cụ thể hơn, mẫu của kênh S2 được nhân với nghịch đảo của hệ số α_1 (được nhận dạng là "coeff[1,2]") của ma trận P_0^{-1} , mẫu của kênh S3 được nhân với nghịch đảo của hệ số α_2 (được nhận dạng là "coeff[1,3]") của ma trận P_0^{-1} , và mẫu của kênh S4 được nhân với nghịch đảo của hệ số α_3 (được nhận dạng là "coeff[1,4]") của ma trận P_0^{-1} , các tích được cộng lại và sau đó được lượng tử hóa, và tổng được lượng tử hóa sau đó được trừ đi từ mẫu tương ứng của kênh S1. Tương tự, mẫu của kênh S1 được nhân với nghịch đảo của hệ số α_0 (được nhận dạng là "coeff[2,1]") của ma trận P_1^{-1} , mẫu của kênh S3 được nhân với nghịch đảo của hệ số α_2 (được nhận dạng là "coeff[2,3]") của ma trận P_1^{-1} , và mẫu của kênh S4 được nhân với nghịch đảo của hệ số α_3 (được nhận dạng là "coeff[2,4]") của ma trận P_1^{-1} , các tích được cộng lại và sau đó được lượng tử hóa, và tổng được lượng tử hóa này sau đó được trừ đi từ mẫu tương ứng của kênh S2. Bước lượng tử hóa Q1 của ma trận P_0^{-1} lượng tử hóa đầu ra của phần tử tính tổng mà cộng các tích của các phép nhân (với các hệ số α khác 0 của ma trận P_0^{-1} , mà thường là các giá trị phân số) để tạo ra giá trị lượng tử hóa mà bị trừ từ mẫu của kênh S1 để tạo ra mẫu tương ứng của kênh mã hóa S1'. Bước lượng tử hóa Q2 của ma trận P_1^{-1} lượng tử hóa đầu ra của phần tử tính tổng mà cộng các tích của các phép nhân (với các hệ số α khác 0 của ma trận P_1^{-1} , mà thường là các giá trị phân số) để tạo ra giá trị lượng tử hóa mà bị trừ từ mẫu của kênh S2 để tạo ra mẫu tương ứng của kênh mã hóa S2'. Theo phương án thực hiện thông thường (ví dụ, để thực hiện mã hóa TrueHD), mỗi mẫu của mỗi kênh S1, S2, S3, và S4 bao gồm 24 bit (được biểu thị trên Fig.2A), và đầu ra của mỗi phần tử nhân bao gồm 38 bit (cũng được biểu thị trên Fig.2A), và mỗi bước lượng tử hóa

Q1 và Q2 xuất ra giá trị 24 bit được lượng tử hóa đáp lại mỗi giá trị 38 bit mà được đưa vào từ đó.

Tất nhiên, để mã hóa các kênh S3 và S4, hai ma trận nguyên thủy bổ sung có thể được nối tăng với hai ma trận nguyên thủy (P_0^{-1} và P_1^{-1}) được biểu thị trên Fig.2A.

Trên Fig.2B (thể hiện hệ mạch để giải mã chương trình mã hóa bốn kênh được tạo ra bởi bộ mã hóa trên Fig.2A), ma trận nguyên thủy P_1 (có một hàng gồm bốn hệ số α khác 0, và là nghịch đảo của ma trận P_1^{-1}) hoạt động trên mỗi mẫu của kênh mã hóa S2' (để tạo ra mẫu tương ứng của kênh giải mã S2) bằng cách trộn các mẫu của các kênh S1', S3, và S4 với mẫu liên quan của kênh S2'. Ma trận nguyên thủy thứ hai P_0 (cũng có một hàng gồm bốn hệ số α khác 0, và là nghịch đảo của ma trận P_0^{-1}) hoạt động trên mỗi mẫu của kênh mã hóa S1' (để tạo ra mẫu tương ứng của kênh giải mã S1) bằng cách trộn các mẫu của các kênh S2, S3, và S4 với mẫu liên quan của kênh S1'. Cụ thể là, mẫu của kênh S1' được nhân với hệ số α_0 (được nhận dạng là "coeff[2,1]") của ma trận P_1 , mẫu của kênh S3 được nhân với hệ số α_2 (được nhận dạng là "coeff[2,3]") của ma trận P_1 , và mẫu của kênh S4 được nhân với hệ số α_3 (được nhận dạng là "coeff[2,4]") của ma trận P_1 , các tích được cộng và sau đó được lượng tử hóa, và tổng được lượng tử hóa này sau đó được bổ sung vào mẫu tương ứng của kênh S1'. Tương tự, mẫu của kênh S2' được nhân với hệ số α_1 (được nhận dạng là "coeff[1,2]") của ma trận P_0 , mẫu của kênh S3 được nhân với hệ số α_2 (được nhận dạng là "coeff[1,3]") của ma trận P_0 , và mẫu của kênh S4 được nhân với hệ số α_3 (được nhận dạng là "coeff[1,4]") của ma trận P_0 , các tích được cộng và sau đó được lượng tử hóa, và tổng được lượng tử hóa này sau đó được bổ sung vào mẫu tương ứng của kênh S1'. Bước lượng tử hóa Q2 của ma trận P_1 lượng tử hóa đầu ra của phần tử tính tổng mà cộng các tích của các phép nhân (với các hệ số α khác 0 của ma trận P_1 , các hệ số này thường là các giá trị phân số) để tạo ra giá trị được lượng tử hóa mà được thêm vào mẫu của kênh S2' để tạo ra mẫu tương ứng của kênh được giải mã S2. Bước lượng tử hóa Q1 của ma trận P_0 lượng tử hóa đầu ra của phần tử tính tổng mà cộng các tích của các phép nhân (với các hệ số α khác 0 của ma trận P_0 , mà thông thường là các giá trị phân số) để tạo ra giá trị được lượng tử hóa mà được thêm vào mẫu của kênh S1' để tạo ra mẫu tương ứng của kênh được giải mã S1. Theo phương án thực hiện thông thường (ví dụ, để thực hiện việc giải mã TrueHD), mỗi mẫu của mỗi kênh S1', S2', S3, và S4 bao gồm 24 bit (như được biểu thị trên Fig.2B), và đầu

ra của mỗi phần tử nhân bao gồm 38 bit (cũng được biểu thị trên Fig.2B), và mỗi bước lượng tử hóa Q1 và Q2 xuất ra giá trị 24 bit được lượng tử hóa để đáp lại mỗi giá trị 38 bit mà được đưa nhập vào đó.

Tất nhiên, để giải mã các kênh S3 và S4, hai ma trận nguyên thủy bổ sung có thể được nối với hai ma trận nguyên thủy (P_0 và P_1) được biểu thị trên Fig.2B.

Tầng của ma trận nguyên thủy, ví dụ, tầng của ma trận $N \times N$ nguyên thủy $P_0, P_1, \dots, P_n$ được thực hiện bởi bộ giải mã trên Fig.1, hoạt động trên vector (N mẫu, mỗi trong số này là mẫu của một kênh khác của tập hợp thứ nhất gồm N kênh) có thể thực hiện sự biến đổi tuyến tính bất kỳ của N mẫu thành tập hợp mới gồm N mẫu (ví dụ, nó có thể thực hiện phép biến đổi tuyến tính được thực hiện ở thời điểm t bằng cách nhân các mẫu gồm N kênh của chương trình âm thanh dựa trên đối tượng với các mẫu bổ sung $N \times N$ bất kỳ của ma trận $A(t)$ trong phương trình (1) khi kết xuất các kênh thành N tín hiệu nạp vào loa, trong đó việc biến đổi đạt được bằng cách thao tác trên một kênh vào một thời điểm). Vì vậy, phép nhân của tập hợp gồm N mẫu âm thanh với dãy $N \times N$ ma trận nguyên thủy biểu diễn một tập hợp tình huống chung trong đó tập hợp N mẫu được biến đổi thành tập hợp khác (gồm N mẫu) bằng các phép tuyến tính.

Xem lại phương án thực hiện TrueHD của bộ giải mã 32 trên Fig.1, để duy trì tính thống nhất của cấu trúc bộ giải mã trong TrueHD, các ma trận đầu ra của dòng phụ trộn giảm (P_0^2, P_1^2 trên Fig.1) cũng được thực hiện như các ma trận nguyên thủy mặc dù chúng không cần nghịch đảo được (hoặc có đường chéo đơn vị) vì chúng không liên quan đến việc đạt được tính không mất dữ liệu.

Các ma trận nguyên thủy đầu vào và đầu ra được sử dụng trong bộ mã hóa và bộ giải mã TrueHD phụ thuộc vào mỗi đặc tả trộn giảm cụ thể cần được thực hiện. Chức năng của bộ giải mã TrueHD là áp dụng dãy phù hợp của các ma trận nguyên thủy cho dòng bit âm thanh mã hóa nhận được. Vì vậy, bộ giải mã TrueHD trên Fig.1 giải mã 8 kênh của dòng bit mã hóa (được phân phối bởi hệ thống D), và tạo ra dạng trộn giảm 2 kênh bằng cách áp dụng tầng hai ma trận nguyên thủy đầu ra P_0^2, P_1^2 cho tập hợp con của các kênh của dòng bit được giải mã. Việc thực hiện TrueHD của bộ giải mã 32 trên Fig.1 cũng có thể hoạt động để giải mã 8 kênh của dòng bit mã hóa (được phân phối bởi hệ thống D) để phục hồi chương trình 8 kênh gốc bằng cách áp

dụng tầng tám ma trận nguyên thủy đầu ra $P_0, P_1, \dots, P_n$ cho các kênh của dòng bit mã hóa.

Bộ giải mã TrueHD không có âm thanh gốc (âm thanh này được đưa vào bộ mã hóa) để kiểm tra nhằm xác định xem quá trình tái tạo âm thanh của nó có phải là quá trình tái tạo không mất dữ liệu không (hoặc theo mong muốn khác của bộ mã hóa trong trường hợp trộn giảm). Tuy nhiên, dòng bit được mã hóa chứa “từ kiểm tra” (hoặc kiểm tra không tồn hao) từ này được so sánh với từ tương tự tạo ra ở bộ giải mã từ âm thanh tái tạo để xác định xem quá trình tái tạo có tin cậy không.

Nếu chương trình âm thanh dựa trên đối tượng (ví dụ, bao gồm nhiều hơn tám kênh) được mã hóa bởi bộ mã hóa TrueHD thông thường, thì bộ mã hóa có thể tạo ra các dòng phụ trộn giảm mang các nội dung trình bày tương thích với các thiết bị phát lại kế thừa (ví dụ, nội dung trình bày mà có thể được giải mã thành tín hiệu nạp cho loa được trộn giảm để phát lại trên kênh 7.1 hoặc kênh 5.1 truyền thống hoặc bộ loa truyền thống khác) và dòng phụ trên (biểu thị tất cả các kênh của chương trình đầu vào). Bộ giải mã TrueHD có thể phục hồi chương trình âm thanh dựa trên đối tượng gốc theo cách không mất dữ liệu để kết xuất bởi hệ thống phát lại. Mỗi đặc tả ma trận kết xuất được sử dụng bởi bộ mã hóa trong trường hợp này (ví dụ, để tạo ra dòng phụ trên và mỗi dòng phụ trộn giảm), và vì vậy mỗi ma trận đầu ra được xác định bởi bộ mã hóa, có thể là ma trận kết xuất thay đổi theo thời gian, $A(t)$, ma trận này biến đổi tuyến tính các mẫu kênh của chương trình (ví dụ, để tạo ra dạng trộn giảm kênh 7.1 hoặc kênh 5.1). Tuy nhiên, ma trận $A(t)$ này thường sẽ thay đổi kịp thời vì các đối tượng chuyển động quanh cảnh không gian, và các giới hạn tốc độ bit và xử lý của hệ thống TrueHD thông thường (hoặc hệ thống giải mã thông thường khác) thường sẽ ràng buộc hệ thống để có thể điều tiết tối đa phép xấp xỉ không đổi theo từng phần cho đặc tả ma trận thay đổi liên tục (và nhanh) (tốc độ cập nhật ma trận cao hơn đạt được khi chi phí tốc độ bit tăng lên để truyền chương trình được mã hóa). Để hỗ trợ việc kết xuất các chương trình âm thanh đa kênh dựa trên đối tượng (và các chương trình âm thanh đa kênh khác) với các tín hiệu nạp vào loa biểu thị việc trộn nội dung thay đổi nhanh từ các kênh của các chương trình, các tác giả sáng chế đã nhận ra mong muốn nâng cao các hệ thống thông thường để điều tiết việc lập ma trận nội suy, trong đó các cập nhật ma trận kết xuất ít khi xảy ra và quỹ đạo mong muốn (ví dụ,

dãy các dạng trộn nội dung của các kênh mong muốn của chương trình) giữa các lần cập nhật được xác định bằng tham số.

Bản chất kỹ thuật của sáng chế

Theo các phương án, sáng chế đề xuất phương pháp mã hóa chương trình âm thanh N kênh (ví dụ, chương trình âm thanh dựa trên đối tượng), trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu mã hóa thành M kênh đầu ra (ví dụ, các kênh tương ứng với các kênh loa phát lại) đã được định rõ trong một khoảng thời gian, trong đó M nhỏ hơn hoặc bằng N, phương pháp này bao gồm các bước:

xác định tầng thứ nhất của ma trận nguyên thủy $N \times N$ mà, khi được áp dụng cho các mẫu của N kênh tín hiệu mã hóa, thực hiện phép trộn thứ nhất nội dung âm thanh của N kênh tín hiệu mã hóa thành M kênh đầu ra, trong đó dạng trộn thứ nhất phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$, theo nghĩa là dạng trộn thứ nhất này ít nhất gần bằng $A(t_1)$;

xác định các giá trị nội suy mà, có tầng thứ nhất của ma trận nguyên thủy và hàm nội suy được xác định theo khoảng thời gian con, biểu thị thứ tự dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, sao cho mỗi trong số các tầng của ma trận nguyên thủy được cập nhật này, khi được áp dụng cho các mẫu của N kênh tín hiệu mã hóa, thực hiện phép trộn được cập nhật, gắn với một thời điểm khác trong khoảng thời gian con, của N kênh tín hiệu mã hóa thành M kênh đầu ra, trong đó mỗi dạng trộn được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$ (tốt hơn là, dạng trộn được cập nhật gắn với thời điểm bất kỳ t_3 trong khoảng thời gian con là ít nhất gần bằng $A(t_3)$, nhưng theo một số phương án có thể có sai số giữa dạng trộn được cập nhật gắn với ít nhất một thời điểm trong khoảng thời gian con và giá trị của $A(t)$ ở thời điểm đó); và

tạo ra dòng bit được mã hóa mà biểu thị nội dung âm thanh được mã hóa, các giá trị nội suy, và tầng thứ nhất của ma trận nguyên thủy.

Theo một số phương án, phương pháp bao gồm bước tạo ra nội dung âm thanh mã hóa bằng cách thực hiện các phép toán ma trận trên các mẫu của N kênh của chương trình (ví dụ, bao gồm bước áp dụng dãy tầng của ma trận cho các mẫu, trong

đó mỗi tầng của ma trận trong dãy là tầng của ma trận nguyên thủy, và dãy tầng ma trận bao gồm tầng của ma trận nghịch đảo thứ nhất là tầng của dạng nghịch đảo của các ma trận nguyên thủy của dãy thứ nhất).

Theo một số phương án, mỗi ma trận nguyên thủy là ma trận nguyên thủy dạng đơn vị. Theo một số phương án trong đó $N = M$, phương pháp này còn bao gồm bước phục hồi không mất dữ liệu N kênh của chương trình bằng cách xử lý dòng bit mã hóa, bao gồm bước thực hiện phép nội suy để xác định dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, từ các giá trị nội suy, tầng thứ nhất của ma trận nguyên thủy, và hàm nội suy. Dòng bit được mã hóa có thể biểu thị (ví dụ, có thể bao gồm dữ liệu biểu thị) hàm nội suy, hoặc nếu không thì hàm nội suy có thể được cung cấp cho bộ giải mã.

Theo một số phương án trong đó $N = M$, phương pháp này cũng bao gồm các bước: phân phối dòng bit được mã hóa cho bộ giải mã được tạo cấu hình để thực hiện hàm nội suy, và xử lý dòng bit được mã hóa ở bộ giải mã để phục hồi không mất dữ liệu N kênh của chương trình, bao gồm bước thực hiện phép nội suy để xác định dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, từ các giá trị nội suy, tầng thứ nhất của ma trận nguyên thủy, và hàm nội suy.

Theo một số phương án, chương trình là chương trình âm thanh dựa trên đối tượng bao gồm ít nhất một kênh đối tượng và dữ liệu vị trí biểu thị quỹ đạo của ít nhất một đối tượng. Ma trận trộn thay đổi theo thời gian, $A(t)$, có thể được xác định từ dữ liệu vị trí (hoặc từ dữ liệu bao gồm dữ liệu vị trí).

Theo một số phương án, tầng thứ nhất của ma trận nguyên thủy là ma trận hạt giống nguyên thủy, và các giá trị nội suy biểu thị ma trận delta hạt giống cho ma trận hạt giống nguyên thủy.

Theo một số phương án, ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, của nội dung âm thanh hoặc nội dung được mã hóa của chương trình thành $M1$ kênh loa cũng đã được định rõ trong một khoảng thời gian, trong đó $M1$ là số nguyên nhỏ hơn M , và phương pháp này bao gồm các bước:

xác định tầng thứ hai của ma trận nguyên thủy $M1 \times M1$ mà, khi được áp dụng vào các mẫu của $M1$ kênh gồm nội dung âm thanh hoặc nội dung được mã hóa, thực hiện phép trộn giảm nội dung âm thanh của chương trình thành $M1$ kênh loa, trong đó

dạng trộn giảm phù hợp này với ma trận trộn thay đổi theo thời gian, $A_2(t)$, theo nghĩa là dạng trộn giảm ít nhất gần bằng $A_2(t_1)$;

xác định các giá trị nội suy bổ sung mà, với tầng thứ hai của ma trận nguyên thủy $M1 \times M1$ và hàm nội suy thứ hai được xác định theo khoảng thời gian con, biểu thị dãy tầng của ma trận nguyên thủy được cập nhật $M1 \times M1$, sao cho mỗi tầng của ma trận nguyên thủy được cập nhật $M1 \times M1$, khi được áp dụng cho các mẫu của $M1$ kênh của nội dung âm thanh hoặc nội dung được mã hóa, thực hiện phép trộn giảm được cập nhật, gắn với một thời điểm khác trong khoảng thời gian con, nội dung âm thanh của chương trình thành $M1$ kênh loa, trong đó mỗi dạng trộn giảm được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian, $A_2(t)$, và trong đó dòng bit được mã hóa biểu thị các giá trị nội suy bổ sung và tầng thứ hai của ma trận nguyên thủy $M1 \times M1$. Dòng bit được mã hóa có thể biểu thị (ví dụ, có thể bao gồm biểu thị dữ liệu) của hàm nội suy thứ hai, hoặc nếu không thì hàm nội suy thứ hai có thể được cung cấp cho bộ giải mã. Ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, là ma trận trộn giảm nội dung âm thanh hoặc nội dung được mã hóa của chương trình theo nghĩa nó là dạng trộn giảm nội dung âm thanh của chương trình gốc, hoặc dạng trộn giảm nội dung âm thanh được mã hóa của dòng bit được mã hóa, hoặc dạng trộn giảm của phiên bản được giải mã một phần của nội dung âm thanh được mã hóa của dòng bit được mã hóa, hoặc nếu không thì là dạng trộn giảm của âm thanh được mã hóa (ví dụ, được giải mã một phần) biểu thị nội dung âm thanh của chương trình. Sự thay đổi theo thời gian trong đặc tả ma trận trộn giảm $A_2(t)$ có thể là do (ít nhất một phần) sự tăng đến hoặc thoát khỏi sự bảo vệ kẹp của dạng trộn giảm được định rõ.

Theo các phương án thứ hai, sáng chế đề xuất phương pháp phục hồi M kênh của chương trình âm thanh đa kênh (ví dụ, chương trình âm thanh dựa trên đối tượng), trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, phương pháp này bao gồm các bước:

thu được dòng bit được mã hóa mà biểu thị của nội dung âm thanh được mã hóa, các giá trị nội suy, và tầng thứ nhất của ma trận nguyên thủy $N \times N$; và

thực hiện phép nội suy để xác định dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, từ các giá trị nội suy, tầng thứ nhất của ma trận nguyên thủy, và hàm nội suy theo khoảng thời gian con, trong đó

tầng thứ nhất của ma trận nguyên thủy $N \times N$, khi được áp dụng cho các mẫu gồm N kênh tín hiệu được mã hóa của nội dung âm thanh được mã hóa, thực hiện phép trộn thứ nhất nội dung âm thanh của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó dạng trộn thứ nhất này phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$, theo nghĩa là dạng trộn thứ nhất ít nhất gần bằng $A(t_1)$, và các giá trị nội suy, với tầng thứ nhất của ma trận nguyên thủy, và hàm nội suy, biểu thị dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, sao cho mỗi trong số các tầng của ma trận nguyên thủy được cập nhật, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa của nội dung âm thanh được mã hóa, thực hiện phép trộn được cập nhật, gắn với một thời điểm khác trong khoảng thời gian con, của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó mỗi dạng trộn được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$ (tốt hơn là, dạng trộn được cập nhật này gắn với thời điểm bất kỳ t_3 trong khoảng thời gian con ít nhất gần bằng $A(t_3)$, nhưng trong một số phương án có thể có sai số giữa dạng trộn được cập nhật gắn với ít nhất một thời điểm trong khoảng thời gian con và giá trị của $A(t)$ ở thời điểm đó).

Theo một số phương án, nội dung âm thanh được mã hóa đã được tạo ra bằng cách thực hiện các phép toán ma trận trên các mẫu của kênh N của chương trình, bao gồm áp dụng dãy các tầng của ma trận cho các mẫu, trong đó mỗi tầng của ma trận trong dãy là tầng của ma trận nguyên thủy, và dãy các tầng của ma trận bao gồm tầng của ma trận nghịch đảo thứ nhất mà là dãy nghịch đảo của các ma trận nguyên thủy thuộc tầng thứ nhất.

Các kênh chương trình âm thanh mà được phục hồi (ví dụ, được phục hồi không mất dữ liệu) theo các phương án này từ dòng bit được mã hóa có thể là dạng trộn giảm của nội dung âm thanh của chương trình âm thanh đầu vào X kênh (trong đó X là số nguyên tùy ý và N nhỏ hơn X) mà đã được tạo ra từ chương trình âm thanh đầu vào X kênh bằng cách thực hiện các phép toán ma trận trên chương trình âm thanh đầu vào X kênh, nhờ đó xác định nội dung âm thanh được mã hóa của dòng bit được mã hóa.

Theo một số phương án trong lớp thứ hai, mỗi trong số các ma trận nguyên thủy là ma trận nguyên thủy dạng đơn vị.

Theo một số phương án trong lớp thứ hai, ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, của chương trình N kênh đến M1 kênh loa đã được định rõ trong một khoảng thời gian, và ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, của nội dung âm thanh hoặc nội dung được mã hóa của chương trình đến M kênh loa cũng đã được định rõ trong một khoảng thời gian. Phương pháp bao gồm các bước:

nhận tầng thứ hai của ma trận nguyên thủy $M1 \times M1$ và tập hợp thứ hai của các giá trị nội suy;

áp dụng tầng thứ hai của ma trận nguyên thủy $M1 \times M1$ cho các mẫu của M1 kênh của nội dung âm thanh được mã hóa để thực hiện phép trộn giảm của chương trình N kênh thành M1 kênh loa, trong đó dạng trộn giảm này phù hợp với ma trận trộn thay đổi theo thời gian, $A_2(t)$, theo nghĩa là dạng trộn giảm ít nhất gần bằng $A_2(t_1)$;

áp dụng tập hợp thứ hai các giá trị nội suy, tầng thứ hai của ma trận nguyên thủy $M1 \times M1$ và hàm nội suy thứ hai được xác định trên khoảng thời gian con để thu được dãy các tầng của ma trận nguyên thủy $M1 \times M1$ được cập nhật; và

áp dụng ma trận nguyên thủy $M1 \times M1$ được cập nhật cho các mẫu của M1 kênh nội dung được mã hóa để thực hiện ít nhất một dạng trộn giảm được cập nhật của chương trình N kênh, gắn với một thời điểm khác trong một khoảng thời gian con, trong đó mỗi dạng trộn giảm được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian, $A_2(t)$.

Theo một số phương án, sáng chế đề xuất phương pháp kết xuất chương trình âm thanh đa kênh, bao gồm các bước tạo ra tập hợp ma trận hạt giống (ví dụ, ma trận hạt giống đơn, hoặc tập hợp của ít nhất hai ma trận hạt giống, tương ứng với thời gian trong chương trình âm thanh) cho bộ giải mã, và thực hiện phép nội suy trên tập hợp ma trận hạt (tập hợp này được kết hợp với thời gian trong chương trình âm thanh) để xác định tập hợp ma trận kết xuất được nội suy (ma trận kết xuất được nội suy đơn, hoặc tập hợp của ít nhất hai ma trận kết xuất được nội suy, tương ứng với lần sau trong suốt chương trình âm thanh) dùng để kết xuất các kênh của chương trình.

Theo một số phương án, ma trận hạt giống nguyên thủy và ma trận đenta hạt giống (hoặc tập hợp các ma trận hạt giống nguyên thủy và các ma trận đenta hạt giống) thỉnh thoảng được phân phối (ví dụ, ít khi) cho bộ giải mã. Bộ giải mã cập nhật mỗi ma trận hạt giống nguyên thủy (tương ứng với thời điểm, t_1) bằng cách tạo ra ma trận nguyên thủy được nội suy (trong thời điểm, t , muộn hơn t_1) theo phương án của sáng chế từ ma trận hạt giống nguyên thủy và ma trận đenta hạt giống tương ứng, và hàm nội suy $f(t)$. Dữ liệu biểu thị hàm nội suy có thể được phân phối các ma trận dạng hạt hoặc hàm nội suy có thể được xác định trước (ví dụ, được biết trước bởi cả bộ mã hóa và bộ giải mã). Theo cách khác, ma trận hạt giống nguyên thủy (hoặc tập hợp các ma trận hạt giống nguyên thủy) thỉnh thoảng được phân phối (ví dụ, ít khi) cho bộ giải mã. Bộ giải mã cập nhật mỗi ma trận hạt giống nguyên thủy (tương ứng với thời điểm, t_1) bằng cách tạo ra ma trận nguyên thủy được nội suy (trong thời điểm, t , muộn hơn t_1) theo phương án của sáng chế từ ma trận hạt giống nguyên thủy và hàm nội suy $f(t)$, ví dụ, không nhất thiết sử dụng ma trận đenta hạt giống mà tương ứng với ma trận hạt giống nguyên thủy. Dữ liệu biểu thị hàm nội suy có thể được phân phối cho ma trận (hoặc các ma trận) nguyên thủy hạt hoặc chức năng có thể được xác định trước (ví dụ, biết trước bởi cả hai bộ mã hóa và bộ giải mã).

Theo các phương án thông thường, mỗi ma trận nguyên thủy là ma trận nguyên thủy dạng đơn vị. Trong trường hợp này, nghịch đảo của ma trận nguyên thủy được xác định một cách đơn giản bằng cách đảo ngược (nhân với -1) mỗi hệ số không tầm thường của nó (mỗi hệ số α của nó). Điều này cho phép các phép nghịch đảo của các ma trận nguyên thủy (mà được áp dụng bởi bộ mã hóa để mã hóa dòng bit) cần được xác định một cách hiệu quả hơn, và cho phép sử dụng kỹ thuật xử lý có độ chính xác hữu hạn (ví dụ, các mạch có độ chính xác hữu hạn) để thực hiện các phép nhân ma trận được yêu cầu ở bộ mã hóa và bộ giải mã.

Theo các khía cạnh, sáng chế đề xuất hệ thống hoặc thiết bị (ví dụ, bộ mã hóa hoặc bộ giải mã) được tạo cấu hình (ví dụ, được lập trình) để thực hiện phương pháp theo phương án bất kỳ của sáng chế, hệ thống hoặc thiết bị bao gồm bộ đệm mà lưu trữ (ví dụ, theo cách bất biến) ít nhất một khung hoặc phần khác của chương trình âm thanh được mã hóa được tạo ra bởi phương án bất kỳ của phương pháp theo sáng chế hoặc các bước của chúng, và vật ghi đọc được bởi máy tính (ví dụ, đĩa) mà lưu trữ mã (ví dụ, theo cách bất biến) để thực hiện phương án bất kỳ của phương pháp theo sáng

chế hoặc các bước của chúng. Ví dụ, hệ thống theo sáng chế có thể là hoặc bao gồm bộ xử lý chuyên dụng lập trình được, bộ xử lý tín hiệu số, hoặc bộ vi xử lý, được lập trình với phần cứng hoặc phần sụn và/hoặc theo cách khác được tạo cấu hình để thực hiện bất kỳ trong số các phép toán trên dữ liệu, bao gồm phương án của phương pháp theo sáng chế hoặc các bước của chúng. Bộ xử lý chuyên dụng này có thể là hoặc bao gồm hệ thống máy tính bao gồm thiết bị đầu vào, bộ nhớ, và hệ mạch xử lý được lập trình (và/hoặc theo cách khác được tạo cấu hình) để thực hiện phương án của phương pháp theo sáng chế (hoặc các bước của phương pháp này) đáp lại dữ liệu được xác nhận ở đó.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối minh họa các phần tử của hệ thống thông thường bao gồm bộ mã hóa, hệ thống phụ phân phối, và bộ giải mã.

Fig.2A là sơ đồ minh họa hệ mạch của bộ mã hóa thông thường để thực hiện các phép toán lập ma trận không mất dữ liệu thông qua các ma trận nguyên thủy được thực hiện với thuật toán có độ chính xác hữu hạn.

Fig.2B là sơ đồ minh họa hệ mạch của bộ giải mã thông thường để thực hiện các phép toán lập ma trận không mất dữ liệu thông qua các ma trận nguyên thủy được thực hiện với thuật toán có độ chính xác hữu hạn.

Fig.3 là sơ đồ khối minh họa hệ mạch được sử dụng theo một phương án của sáng chế để áp dụng ma trận nguyên thủy 4×4 (được thực hiện với thuật toán có độ chính xác hữu hạn) cho bốn kênh của chương trình âm thanh. Ma trận nguyên thủy là ma trận hạt giống nguyên thủy, mà một hàng không tầm thường của nó bao gồm các phần tử α_0 , α_1 , α_2 , và α_3 .

Fig.4 là sơ đồ khối minh họa hệ mạch được sử dụng theo một phương án của sáng chế để áp dụng ma trận nguyên thủy 3×3 (được thực hiện với thuật toán có độ chính xác hữu hạn) cho ba kênh của chương trình âm thanh. Ma trận nguyên thủy là ma trận nguyên thủy được nội suy, được tạo ra từ ma trận hạt giống nguyên thủy $P_k(t_1)$ mà một hàng không tầm thường của nó bao gồm các phần tử α_0 , α_1 , và α_2 , và ma trận delta hạt giống $\Delta_k^{(t_1)}$ mà hàng không tầm thường của nó bao gồm các phần tử δ_0 , δ_1 , ..., δ_{N-1} , và hàm nội suy $f(t)$.

Fig.5 là sơ đồ khối minh họa một phương án của hệ thống bao gồm bộ mã hóa, hệ thống phụ phân phối, và bộ giải mã theo sáng chế.

Fig.6 là sơ đồ khối minh họa một phương án khác của hệ thống bao gồm bộ mã hóa, hệ thống phụ phân phối, và bộ giải mã theo sáng chế.

Fig.7 là đồ thị minh họa tổng các sai số bình phương giữa đặc điểm thu được và đặc điểm thực ở các thời điểm khác nhau, t, nhờ sử dụng các ma trận nguyên thủy được nội suy (đường cong được đánh dấu là “Lập ma trận nội suy”) và các ma trận nguyên thủy (không được nội suy) không đổi theo từng phần (đường cong được đánh dấu là “Lập ma trận không nội suy”).

Mô tả chi tiết sáng chế

Trong toàn bộ bản mô tả này, bao gồm cả phần yêu cầu bảo hộ, thuật ngữ thực hiện thao tác “trên” tín hiệu hoặc dữ liệu (ví dụ, lọc, định tỷ lệ, biến đổi, hoặc áp dụng độ lợi cho, tín hiệu hoặc dữ liệu) được dùng với nghĩa rộng để chỉ việc thực hiện thao tác trực tiếp trên tín hiệu hoặc dữ liệu, hoặc trên phiên bản đã xử lý của tín hiệu hoặc dữ liệu (ví dụ, trên phiên bản của tín hiệu mà trải qua quá trình lọc sơ bộ hoặc tiền xử lý trước khi thực hiện thao tác trên đó).

Trong toàn bộ bản mô tả này bao gồm cả phần yêu cầu bảo hộ, thuật ngữ “hệ thống” được sử dụng theo nghĩa rộng để chỉ thiết bị, hệ thống, hoặc hệ thống phụ. Ví dụ, hệ thống phụ mà có bộ giải mã có thể được gọi là hệ thống bộ giải mã, và hệ thống bao gồm hệ thống phụ này (ví dụ, hệ thống mà tạo ra Y tín hiệu đầu ra đáp ứng nhiều đầu vào, trong đó hệ thống phụ tạo ra M đầu vào và $Y - M$ đầu vào khác được nhận từ nguồn bên ngoài) cũng có thể được gọi là hệ thống bộ giải mã.

Trong toàn bộ bản mô tả này bao gồm cả phần yêu cầu bảo hộ, thuật ngữ “bộ xử lý” được sử dụng theo nghĩa rộng để chỉ hệ thống hoặc thiết bị lập trình được hoặc theo cách khác cấu hình được (ví dụ, với phần mềm hoặc phần sụn) để thực hiện các thao tác trên dữ liệu (ví dụ, âm thanh, hoặc video hoặc dữ liệu ảnh khác). Ví dụ, về bộ xử lý bao gồm mảng công lập trình được bằng trường (hoặc mạch tích hợp hoặc bộ vi mạch cấu hình được khác), bộ xử lý tín hiệu số được lập trình và/hoặc theo cách khác được tạo cấu hình để thực hiện xử lý kiểu đường ống trên âm thanh hoặc dữ liệu âm thanh khác, bộ xử lý chuyên dụng lập trình được hoặc máy tính, và chip xử lý lập trình được hoặc bộ vi mạch.

Trong toàn bộ bản mô tả này bao gồm cả phần yêu cầu bảo hộ, thuật ngữ “siêu dữ liệu” chỉ dữ liệu riêng biệt và khác với dữ liệu âm thanh tương ứng (nội dung âm thanh của dòng bit mà cũng chứa siêu dữ liệu). Siêu dữ liệu được gắn với dữ liệu âm thanh, và biểu thị ít nhất một dấu hiệu hoặc đặc điểm của dữ liệu âm thanh (ví dụ, (các) kiểu xử lý nào đã được thực hiện, hoặc nên được thực hiện, đối với dữ liệu âm thanh, hoặc quỹ đạo của đối tượng được biểu thị bởi dữ liệu âm thanh). Sự kết hợp của siêu dữ liệu với dữ liệu âm thanh là sự kết hợp đồng bộ theo thời gian. Vì vậy, siêu dữ liệu hiện thời (được nhận hoặc cập nhật gần nhất) có thể biểu thị rằng dữ liệu âm thanh tương ứng có đồng thời dấu hiệu được biểu thị và/hoặc bao gồm kết quả của kiểu xử lý dữ liệu âm thanh được biểu thị.

Trong toàn bộ bản mô tả này bao gồm cả phần yêu cầu bảo hộ, thuật ngữ “nối” hoặc “được nối” được dùng để chỉ kết nối trực tiếp hoặc gián tiếp. Vì vậy, nếu thiết bị thứ nhất nối với thiết bị thứ hai, thì kết nối đó có thể là kết nối trực tiếp, hoặc kết nối gián tiếp qua các thiết bị và kết nối khác.

Trong toàn bộ bản mô tả này bao gồm cả phần yêu cầu bảo hộ, các thuật ngữ sau đây có các nghĩa sau:

loa và loa phóng thanh được sử dụng một cách đồng nghĩa để chỉ bộ chuyển đổi phát ra âm thanh. Định nghĩa này bao gồm các loa được sử dụng như nhiều bộ chuyển đổi (ví dụ, loa trầm và loa cao tần);

tín hiệu nạp vào loa: tín hiệu âm thanh cần được áp dụng trực tiếp cho loa, hoặc tín hiệu âm thanh sẽ được áp dụng cho hàng loạt bộ khuếch đại và loa;

kênh (hoặc “kênh âm thanh”): tín hiệu âm thanh đơn âm. Tín hiệu này thường có thể được kết xuất theo cách tương đương với việc áp dụng tín hiệu trực tiếp vào loa ở vị trí mong muốn hoặc danh định. Vị trí mong muốn có thể là tĩnh, thường là trường hợp có các loa vật lý, hoặc động;

Chương trình âm thanh: tập hợp của một hoặc nhiều kênh âm thanh (ít nhất một kênh loa và/hoặc ít nhất một kênh đối tượng) và tùy ý cũng bao gồm siêu dữ liệu gắn kèm (ví dụ, siêu dữ liệu mô tả biểu diễn âm thanh không gian mong muốn);

kênh loa (hoặc “kênh nạp vào loa”): kênh âm thanh mà được kết hợp với loa được đặt tên (ở vị trí mong muốn hoặc danh định), hoặc với vùng loa được đặt tên trong cấu hình loa xác định. Kênh loa được kết xuất theo cách tương đương với việc

áp dụng tín hiệu âm thanh trực tiếp vào loa được đặt tên (ở vị trí mong muốn hoặc danh định) hoặc vào loa trong vùng loa được đặt tên;

kênh đối tượng: kênh âm thanh biểu thị âm thanh được phát ra bởi nguồn âm thanh (đôi khi được gọi là “đối tượng” âm thanh). Thông thường, kênh đối tượng xác định phần mô tả nguồn âm thanh có thông số (ví dụ, siêu dữ liệu biểu thị phần mô tả nguồn âm thanh có thông số được bao gồm trong hoặc được cấp kênh đối tượng). Phần mô tả nguồn có thể xác định âm thanh phát ra bởi nguồn (là hàm thời gian), vị trí biểu hiện (ví dụ, các tọa độ không gian ba chiều) của của nguồn là hàm thời gian, và tùy ý ít nhất một thông số bổ sung (ví dụ, kích thước hoặc độ rộng nguồn biểu hiện) đặc trưng cho nguồn; và

chương trình âm thanh dựa trên đối tượng: chương trình âm thanh bao gồm tập hợp một hoặc nhiều kênh đối tượng (và tùy ý cũng bao gồm ít nhất một kênh loa) và tùy ý còn bao gồm siêu dữ liệu gắn kèm (ví dụ, siêu dữ liệu biểu thị quỹ đạo của đối tượng âm thanh phát ra âm thanh được biểu thị bởi kênh đối tượng, hoặc siêu dữ liệu biểu thị biểu diễn âm thanh không gian mong muốn của âm thanh được biểu thị bởi kênh đối tượng, hoặc siêu dữ liệu biểu thị sự nhận biết ít nhất một đối tượng âm thanh mà là nguồn âm thanh được biểu thị bởi kênh đối tượng).

Các ví dụ về các phương án của sáng chế sẽ được mô tả dựa vào các Fig.3, Fig.4, Fig.5, và Fig.6.

Fig.5 là sơ đồ khối của hệ thống xử lý dữ liệu âm thanh theo một phương án của sáng chế, hệ thống này bao gồm bộ mã hóa 40 (bộ mã hóa theo một phương án của sáng chế), hệ thống phụ phân phối 41 (hệ thống này có thể giống hệ thống phụ phân phối 31 trên Fig.1), và bộ giải mã 42 (bộ giải mã theo một phương án của sáng chế), được nối với nhau như thể hiện. Mặc dù hệ thống phụ 42 được gọi trong bản mô tả này là “bộ giải mã” nhưng cần hiểu rằng hệ thống này có thể được sử dụng như hệ thống phát lại bao gồm hệ thống phụ giải mã (được tạo cấu hình để phân tích và giải mã dòng bit biểu thị chương trình âm thanh đa kênh được mã hóa) và các hệ thống phụ khác được tạo cấu hình để thực hiện việc kết xuất và ít nhất một số bước phát lại đầu ra của hệ thống phụ giải mã. Một số phương án của sáng chế là các bộ giải mã mà không được tạo cấu hình để thực hiện kết xuất và/hoặc phát lại (và thường được sử dụng với hệ thống kết xuất và/hoặc phát lại riêng). Một số phương án của sáng chế là các hệ thống phát lại (ví dụ, hệ thống phát lại bao gồm hệ thống phụ giải mã và các

hệ thống phụ khác được tạo cấu hình để thực hiện kết xuất và ít nhất một số bước phát lại đầu ra của hệ thống phụ giải mã.

Trong hệ thống trên Fig.5, bộ mã hóa 40 được tạo cấu hình để mã hóa chương trình âm thanh 8 kênh (ví dụ, tập hợp tín hiệu nạp vào loa 7.1 truyền thống) là dòng bit được mã hóa bao gồm hai dòng phụ, và bộ giải mã 42 được tạo cấu hình để giải mã dòng bit được mã hóa để kết xuất chương trình gốc 8 kênh (theo cách không mất dữ liệu) hoặc dạng trộn giảm 2 kênh của chương trình gốc 8 kênh. Bộ mã hóa 40 được nối và được tạo cấu hình để tạo ra dòng bit được mã hóa và để xác nhận dòng bit được mã hóa với hệ thống phân phối 41.

Hệ thống phân phối 41 được nối và được tạo cấu hình để phân phối (ví dụ, lưu trữ và/hoặc truyền) dòng bit được mã hóa cho bộ giải mã 42. Theo một số phương án, hệ thống 41 thực hiện phân phối (ví dụ, truyền) chương trình âm thanh đa kênh được mã hóa trên hệ thống phát rộng hoặc mạng (ví dụ, internet) cho bộ giải mã 42. Theo một số phương án, hệ thống 41 lưu trữ chương trình âm thanh đa kênh được mã hóa vào vật ghi lưu trữ (ví dụ, đĩa hoặc bộ đĩa), và bộ giải mã 42 được tạo cấu hình để đọc chương trình từ vật ghi lưu trữ này.

Khối có ký hiệu “InvChAssign1” trong bộ mã hóa 40 được tạo cấu hình để thực hiện hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) trên các kênh của chương trình đầu vào. Sau đó, các kênh được hoán vị trải qua quá trình mã hóa ở bước 43, bước này xuất ra tám kênh tín hiệu được mã hóa. Các kênh tín hiệu được mã hóa có thể (nhưng không cần thiết) tương ứng với các kênh loa phát lại. Các kênh tín hiệu được mã hóa đôi khi được gọi là các kênh “nội bộ” vì bộ giải mã (và/hoặc hệ thống kết xuất) thường giải mã và kết xuất nội dung của các kênh tín hiệu được mã hóa để phục hồi âm thanh đầu vào, sao cho các kênh tín hiệu được mã hóa là “nội bộ” đối với hệ thống mã hóa/giải mã. Bước mã hóa được thực hiện ở bước 43 tương đương với bước nhân mỗi tập hợp mẫu của các kênh được hoán vị với ma trận mã hóa (được thực hiện dưới dạng một tầng các phép nhân ma trận, được xác định là $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$).

Mặc dù n có thể bằng 7 trong một phương án làm ví dụ, nhưng trong phương án này và các phương án biến đổi khác chương trình âm thanh đầu vào bao gồm số lượng (N hoặc X) kênh tùy ý, trong đó N (hoặc X) là số nguyên bất kỳ lớn hơn một,

và n trên Fig.5 có thể là $n = N-1$ (hoặc $n = X-1$ hoặc giá trị khác). Theo các phương án khác, bộ mã hóa được tạo cấu hình để mã hóa chương trình âm thanh đa kênh dưới dạng dòng bit được mã hóa bao gồm một số dòng phụ, và bộ giải mã được tạo cấu hình để giải mã dòng bit được mã hóa để kết xuất chương trình đa kênh gốc (theo cách không mất dữ liệu) hoặc một hoặc nhiều dạng trộn giảm của chương trình đa kênh gốc. Ví dụ, bước mã hóa (tương ứng với bước 43) theo phương án thay thế này có thể áp dụng tầng của ma trận nguyên thủy $N \times N$ cho các mẫu của các kênh của chương trình, để tạo ra N kênh tín hiệu được mã hóa mà có thể được biến đổi thành dạng trộn thứ nhất của M kênh đầu ra, trong đó dạng trộn thứ nhất này phù hợp với ma trận trộn thay đổi theo thời gian $A(t)$, được định rõ trong một khoảng thời gian, có nghĩa là dạng trộn thứ nhất ít nhất gần bằng $A(t_1)$, ở đó t_1 là một thời điểm trong khoảng thời gian. Bộ giải mã có thể tạo ra M kênh đầu ra bằng cách áp dụng tầng của ma trận nguyên thủy $N \times N$ nhận được là một phần của nội dung âm thanh được mã hóa. Theo một phương án thay thế, bộ mã hóa cũng có thể tạo ra tầng thứ hai của ma trận nguyên thủy $M_1 \times M_1$ (trong đó M_1 là số nguyên nhỏ hơn N), mà cũng được bao gồm trong nội dung âm thanh được mã hóa. Bộ giải mã có thể áp dụng tầng thứ hai này trên M_1 kênh tín hiệu được mã hóa để thực hiện phép trộn giảm chương trình N kênh thành M_1 kênh loa, trong đó dạng trộn giảm này phù hợp với ma trận trộn thay đổi theo thời gian khác, $A_2(t)$, theo nghĩa là dạng trộn giảm này ít nhất gần bằng $A_2(t_1)$. Theo một phương án thay thế, bộ mã hóa cũng sẽ tạo ra các giá trị nội suy (theo phương án bất kỳ của sáng chế) và bao gồm các giá trị nội suy trong dòng bit được mã hóa xuất ra từ bộ mã hóa, để dùng bởi bộ giải mã để giải mã và kết xuất nội dung của dòng bit được mã hóa theo ma trận trộn thay đổi theo thời gian, $A(t)$, và/hoặc để giải mã và kết xuất dạng trộn giảm của nội dung của dòng bit được mã hóa theo ma trận trộn thay đổi theo thời gian, $A_2(t)$.

Phần mô tả trên Fig.5 sẽ đôi khi đề cập đến tín hiệu đa kênh được nhập vào bộ giải mã là tín hiệu đầu vào 8 kênh đối với trường hợp đặc biệt, nhưng phần mô tả này (với những thay đổi không tầm thường hiển nhiên đối với người có hiểu biết trung bình trong lĩnh vực này) cũng áp dụng cho trường hợp chung bằng cách thay thế các tín hiệu chuẩn cho tín hiệu đầu vào 8 kênh bằng các tín hiệu chuẩn của tín hiệu đầu vào N kênh, thay thế các tín hiệu chuẩn cho các tầng của ma trận nguyên thủy 8 kênh (hoặc 2 kênh) bằng các tín hiệu chuẩn của các ma trận nguyên thủy M kênh (hoặc M_1

kênh), và thay thế các tín hiệu chuẩn cho chế độ phục hồi không mất dữ liệu của tín hiệu đầu vào 8 kênh bằng các tín hiệu chuẩn của chế độ phục hồi không mất dữ liệu của tín hiệu âm thanh M kênh (trong đó tín hiệu âm thanh M kênh đã được xác định bằng cách thực hiện các phép toán ma trận để áp dụng ma trận trộn thay đổi theo thời gian, $A(t)$, cho tín hiệu âm thanh đầu vào N kênh để xác định M kênh tín hiệu được mã hóa).

Tham chiếu bước mã hóa 43 trên Fig.5, mỗi ma trận $P_n^{-1}, \dots, P_1^{-1}$, và P_0^{-1} (và vì vậy đây được áp dụng bởi bước 43) được xác định trong hệ thống phụ 44, và được cập nhật theo từng thời điểm (ít khi) theo ma trận trộn thay đổi theo thời gian được định rõ của N (nếu $N = 8$) kênh của chương trình với N kênh tín hiệu được mã hóa mà đã được định rõ trong một khoảng thời gian.

Hệ thống phụ xác định ma trận 44 được tạo cấu hình để tạo ra dữ liệu biểu thị các hệ số của hai tập hợp ma trận đầu ra (một tập hợp tương ứng với mỗi trong số hai dòng phụ của các kênh được mã hóa). Mỗi tập hợp ma trận đầu ra này được cập nhật theo từng thời điểm, sao cho các hệ số cũng được cập nhật bất cứ lúc nào. Một tập hợp ma trận đầu ra bao gồm hai ma trận kết xuất, $P_0^2(t)$, $P_1^2(t)$, mỗi trong số hai ma trận này là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) kích thước 2×2 , và được dùng để kết xuất dòng phụ thứ nhất (dòng phụ trộn giảm) bao gồm hai trong số các kênh âm thanh được mã hóa của dòng bit được mã hóa (để kết xuất dạng trộn giảm hai kênh của âm thanh đầu vào tám kênh). Tập hợp ma trận đầu ra còn lại gồm tám ma trận kết xuất, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, mỗi trong số chúng là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) kích thước 8×8 , và được dùng để kết xuất dòng phụ thứ hai bao gồm tất cả tám kênh âm thanh được mã hóa của dòng bit được mã hóa (để phục hồi không mất dữ liệu cho chương trình âm thanh đầu vào tám kênh). Đối với mỗi thời điểm, t , tầng của ma trận kết xuất, $P_0^2(t)$, $P_1^2(t)$, có thể được hiểu là ma trận kết xuất cho các kênh của dòng phụ thứ nhất mà kết xuất dạng trộn giảm hai kênh từ hai kênh tín hiệu được mã hóa trong dòng phụ thứ nhất, và tương tự tầng của ma trận kết xuất, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, có thể được hiểu là ma trận kết xuất cho các kênh của dòng phụ thứ hai.

Các hệ số (của mỗi ma trận kết xuất) mà được xuất ra từ hệ thống phụ 44 cho hệ thống phụ đóng gói 45 là siêu dữ liệu biểu thị độ lợi tương đối hoặc tuyệt đối của mỗi kênh cần được bao gồm trong dạng trộn tương ứng của các kênh của chương

trình. Các hệ số của mỗi ma trận kết xuất (đối với một thời điểm trong chương trình) thể hiện mỗi kênh của dạng trộn nên đóng góp bao nhiêu vào dạng trộn của nội dung âm thanh (ở thời điểm tương ứng của dạng trộn được kết xuất) được biểu thị bởi tín hiệu nạp vào loa cho loa của hệ thống phát lại cụ thể.

Tám kênh âm thanh được mã hóa (xuất ra từ bước mã hóa 43), các hệ số ma trận đầu ra (được tạo ra bởi hệ thống phụ 44), và dữ liệu bổ sung thường cũng được đưa vào hệ thống phụ đóng gói 45, hệ thống này tập hợp chúng thành dòng bit được mã hóa mà sau đó được đưa vào hệ thống phân phối 41.

Dòng bit được mã hóa bao gồm dữ liệu biểu thị tám kênh âm thanh được mã hóa, hai tập hợp của các ma trận đầu ra thay đổi theo thời gian (một tập hợp tương ứng với mỗi trong số hai dòng phụ của các kênh được mã hóa), và dữ liệu bổ sung thường cũng được bao gồm (ví dụ, siêu dữ liệu về nội dung âm thanh).

Trong khi hoạt động, bộ mã hóa 40 (và các phương án thay thế của bộ giải mã theo sáng chế, ví dụ, bộ mã hóa 100 trên Fig.6) mã hóa chương trình âm thanh N kênh mà các mẫu của nó tương ứng với khoảng thời gian, trong đó khoảng thời gian bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 . Khi ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, thì bộ mã hóa thực hiện các bước:

xác định tầng thứ nhất của ma trận nguyên thủy $N \times N$ (ví dụ, các ma trận $P_0(t_1)$, $P_1(t_1)$, ..., $P_n(t_1)$, cho thời điểm t_1) mà, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa, thực hiện phép trộn thứ nhất của nội dung âm thanh của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó dạng trộn thứ nhất này phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$, theo nghĩa là dạng trộn thứ nhất này ít nhất gần bằng $A(t_1)$;

tạo ra nội dung âm thanh được mã hóa (ví dụ, đầu ra của bước 43 của bộ mã hóa 40, hoặc đầu ra của bước 103 của bộ mã hóa 100) bằng cách thực hiện các phép toán ma trận trên các mẫu của N kênh của chương trình, bao gồm bước áp dụng dãy các tầng của ma trận cho các mẫu, trong đó mỗi tầng của ma trận trong dãy là tầng của ma trận nguyên thủy, và dãy các tầng của ma trận bao gồm tầng của ma trận nghịch đảo thứ nhất mà là tầng nghịch đảo của các ma trận nguyên thủy của tầng thứ nhất;

xác định các giá trị nội suy (ví dụ, các giá trị nội suy được bao gồm trong đầu ra của bước 44 của bộ mã hóa 40, hoặc trong đầu ra của bước 103 của bộ mã hóa 100) mà, với tầng thứ nhất của ma trận nguyên thủy (ví dụ, được bao gồm trong đầu ra của bước 44 hoặc bước 103) và hàm nội suy được xác định theo khoảng thời gian con, biểu thị dãy của dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, sao cho mỗi trong số các tầng của ma trận nguyên thủy được cập nhật, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa, thực hiện phép trộn được cập nhật, gắn với một thời điểm khác trong khoảng thời gian con, của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó mỗi dạng trộn được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian, $A(t)$. Tốt hơn là, nhưng không nhất thiết (trong tất cả các phương án), mỗi dạng trộn được cập nhật này phù hợp với ma trận trộn thay đổi theo thời gian theo nghĩa là dạng trộn được cập nhật gắn với thời điểm bất kỳ t_3 trong khoảng thời gian con ít nhất gần bằng $A(t_3)$; và

tạo ra dòng bit được mã hóa (ví dụ, đầu ra của bước 45 của bộ mã hóa 40 hoặc đầu ra của bước 104 của bộ mã hóa 100) mà biểu thị của nội dung âm thanh được mã hóa, các giá trị nội suy, và dãy thứ nhất gồm các ma trận nguyên thủy.

Dựa vào bước 44 trên Fig.5, mỗi tập hợp ma trận đầu ra (tập hợp P_0^2, P_1^2 , hoặc tập hợp $P_0, P_1, \dots, P_n$) được cập nhật theo từng thời điểm. Tập hợp thứ nhất gồm các ma trận P_0^2, P_1^2 mà được xuất ra (ở thời điểm thứ nhất, t_1) là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy dạng đơn vị) xác định phép biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong chương trình (ví dụ, trên các mẫu của hai kênh của đầu ra được mã hóa ở bước 43, tương ứng với thời điểm thứ nhất). Tập hợp thứ nhất gồm các ma trận $P_0, P_1, \dots, P_n$ mà được xuất ra (ở thời điểm thứ nhất, t_1) cũng là ma trận hạt giống (được thực hiện dưới dạng tầng của các ma trận nguyên thủy dạng đơn vị) xác định phép biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong chương trình (ví dụ, trên các mẫu của tất cả tám kênh của đầu ra được mã hóa ở bước 43 tương ứng với thời điểm thứ nhất). Mỗi tập hợp cập nhật của các ma trận P_0^2, P_1^2 mà được xuất ra từ bước 44 là ma trận hạt giống được cập nhật (được thực hiện dưới dạng tầng của các ma trận nguyên thủy dạng đơn vị, mà cũng có thể được gọi là tầng của các ma trận hạt giống nguyên thủy dạng đơn vị) xác định phép biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong chương trình

(ví dụ, trên các mẫu gồm hai kênh của đầu ra được mã hóa ở bước 43, tương ứng với thời điểm cập nhật). Mỗi tập hợp được cập nhật của các ma trận $P_0, P_1, \dots, P_n$ mà được xuất ra từ bước 43 cũng là ma trận hạt giống (được thực hiện dưới dạng tầng của ma trận nguyên thủy dạng đơn vị, mà cũng có thể được gọi là tầng của ma trận hạt giống nguyên thủy dạng đơn vị) xác định phép biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong chương trình (ví dụ, trên các mẫu của tất cả tám kênh của đầu ra được mã hóa ở bước 43 tương ứng với thời điểm thứ nhất).

Bước xuất ra 44 cũng xuất ra các giá trị nội suy, mà (với hàm nội suy cho mỗi ma trận hạt giống) cho phép bộ giải mã 42 tạo ra các phiên bản được nội suy của các ma trận hạt (tương ứng với các thời điểm sau thời điểm thứ nhất, t_1 , và giữa các thời điểm cập nhật). Các giá trị nội suy (mà có thể bao gồm dữ liệu biểu thị mỗi hàm nội suy) được bao gồm bởi bước 45 trong dòng bit được mã hóa xuất ra từ bộ mã hóa 40. Sáng chế sẽ mô tả các ví dụ về các giá trị nội suy này dưới đây (các giá trị nội suy có thể bao gồm ma trận delta cho mỗi ma trận hạt giống).

Dựa vào bộ giải mã 42 trên Fig.5, hệ thống phụ phân tích 46 (của bộ giải mã 42) được tạo cấu hình để chấp nhận (đọc hoặc nhận) dòng bit được mã hóa từ hệ thống phân phối 41 và phân tích dòng bit được mã hóa. Hệ thống phụ 46 có thể hoạt động để xác nhận các dòng phụ của dòng bit được mã hóa (bao gồm dòng phụ “thứ nhất” chỉ chứa hai kênh được mã hóa của dòng bit được mã hóa), và các ma trận đầu ra (P_0^2, P_1^2) tương ứng với dòng phụ thứ nhất, với bước nhân ma trận 48 (để xử lý xem dòng phụ nào dẫn đến sự biểu diễn trộn giảm 2 kênh của nội dung của chương trình đầu vào 8 kênh gốc). Hệ thống phụ 46 cũng có thể hoạt động để xác nhận các dòng phụ của dòng bit được mã hóa (dòng phụ “thứ hai” bao gồm tất cả tám kênh được mã hóa của dòng bit được mã hóa), và các ma trận đầu ra tương ứng ($P_0, P_1, \dots, P_n$) với bước nhân ma trận 47 để xử lý dòng phụ nào dẫn đến sự tái tạo không mất dữ liệu đối với chương trình gốc 8 kênh.

Hệ thống phụ phân tích 46 (và hệ thống phụ phân tích 105 trên Fig.6) có thể bao gồm (và/hoặc thực hiện) các công cụ mã hóa và giải mã không mất dữ liệu bổ sung (ví dụ, mã hóa LPC, mã hóa Huffman, và v.v.).

Bước nội suy 60 được liên kết để nhận mỗi ma trận hạt giống cho dòng phụ thứ hai (ví dụ, tập hợp ban đầu gồm ma trận nguyên thủy, $P_0, P_1, \dots, P_n$, cho thời điểm t_1 ,

và mỗi tập hợp các ma trận nguyên thủy được cập nhật, $P_0, P_1, \dots, P_n$) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng được bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy của mỗi ma trận hạt giống. Bước 60 được liên kết và được tạo cấu hình để đi qua (đến bước 47) mỗi ma trận hạt giống này, và tạo ra (và xác nhận với bước 47) các phiên bản được nội suy của mỗi ma trận hạt giống này (mỗi phiên bản được nội suy tương ứng với một thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống tiếp theo).

Bước nội suy 61 được liên kết để nhận mỗi ma trận hạt giống cho dòng con thứ nhất (ví dụ, tập hợp ban đầu gồm các ma trận nguyên thủy, P_0^2 và P_1^2 , cho thời điểm t_1 , và mỗi tập hợp các ma trận nguyên thủy được cập nhật, P_0^2 và P_1^2) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng được bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy của mỗi ma trận hạt giống này. Bước 61 được liên kết và được tạo cấu hình để đi qua (đến bước 48) mỗi ma trận hạt giống này, và tạo ra (và xác nhận với bước 48) các phiên bản được nội suy của mỗi ma trận hạt giống (mỗi phiên bản được nội suy tương ứng với một thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống tiếp theo).

Bước 48 nhân hai mẫu âm thanh của hai kênh (của dòng bit được mã hóa) mà tương ứng với các kênh của dòng phụ thứ nhất với dãy được cập nhật mới nhất của các ma trận P_0^2 và P_1^2 (ví dụ, dãy các phiên bản mới nhất được nội suy gồm các ma trận P_0^2 và P_1^2 được tạo ra bởi bước 61), và mỗi tập hợp kết quả gồm hai mẫu được biến đổi tuyến tính trải qua quá trình hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign0” để tạo ra mỗi cặp mẫu của dạng trộn giảm 2 kênh cần thiết của 8 kênh âm thanh gốc. Tầng của phép toán lập ma trận được thực hiện trong bộ mã hóa 40 và bộ giải mã 42 tương đương với việc áp dụng đặc tả của ma trận trộn giảm mà biến đổi 8 kênh âm thanh đầu vào thành ma trận trộn giảm 2 kênh.

Bước 47 nhân mỗi vector gồm tám mẫu âm thanh (một mẫu từ mỗi trong số toàn bộ tập hợp gồm tám kênh của dòng bit được mã hóa) với tầng được cập nhật mới nhất của các ma trận $P_0, P_1, \dots, P_n$ (ví dụ, tầng các phiên bản nội suy mới nhất gồm các ma

trận $P_0, P_1, \dots, P_n$ được tạo ra bởi bước 60) và mỗi tập hợp kết quả gồm tám mẫu được biến đổi tuyến tính trải qua quá trình hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối được ký hiệu “ChAssign1” để tạo ra mỗi tập hợp gồm tám mẫu của chương trình tám kênh gốc được phục hồi không mất dữ liệu. Để cho âm thanh 8 kênh đầu ra giống âm thanh 8 kênh đầu vào (để đạt được đặc tính “không mất dữ liệu” của hệ thống), các phép toán lập ma trận được thực hiện trong bộ mã hóa 40 cần phải chính xác (bao gồm các tác dụng lượng tử hóa) là dạng nghịch đảo của các phép toán lập ma trận được thực hiện trong bộ giải mã 42 trên dòng phụ thứ hai của dòng bit được mã hóa (ví dụ, mỗi lần nhân ở bước 47 trong bộ giải mã 42 với dãy các ma trận $P_0, P_1, \dots, P_n$). Vì vậy, trên Fig.5, các phép toán lập ma trận ở bước 43 của bộ mã hóa 40 được nhận dạng là tầng của ma trận nghịch đảo của các ma trận $P_0, P_1, \dots, P_n$, trong dãy đối diện được áp dụng ở bước 47 của bộ giải mã 42, tức là: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

Vì vậy, bước 47 (với bước hoán vị, ChAssign1) là hệ thống phụ nhân ma trận được liên kết và tạo cấu hình để áp dụng tuần tự mỗi tầng của ma trận nguyên thủy xuất ra từ bước nội suy 60 cho nội dung âm thanh được mã hóa được trích từ dòng bit được mã hóa, để phục hồi không mất dữ liệu N kênh của ít nhất một đoạn của chương trình âm thanh đa kênh mà được mã hóa bởi bộ mã hóa 40.

Giai đoạn hoán vị ChAssign1 của bộ giải mã 42 áp dụng cho đầu ra của giai đoạn 47 bước nghịch đảo của hoán vị kênh được áp dụng bởi bộ mã hóa 40 (ví dụ, ma trận hoán vị được biểu diễn bởi giai đoạn “ChAssign1” của bộ giải mã 42 là nghịch đảo của ma trận được biểu diễn bởi phần tử “InvChAssign1” của bộ mã hóa 40).

Trong các biến thể trên các hệ thống phụ 40 và 42 của hệ thống thể hiện trên Fig.5, một hoặc nhiều trong số các phần tử được bỏ qua hoặc các đơn vị xử lý dữ liệu âm thanh bổ sung được bao gồm.

Trong các biến thể trong phương án được mô tả của bộ giải mã 42, bộ giải mã của sáng chế được tạo cấu hình để thực hiện phục hồi không mất dữ liệu N kênh gồm nội dung âm thanh được mã hóa từ dòng bit được mã hóa biểu thị N kênh tín hiệu được mã hóa, trong đó N kênh gồm nội dung âm thanh chính là dạng trộn giảm của nội dung âm thanh của chương trình âm thanh đầu vào X kênh (trong đó X là số nguyên tùy ý và N là nhỏ hơn X), được tạo ra bằng cách thực hiện các phép toán ma

trên chương trình âm thanh đầu vào X kênh để áp dụng ma trận trộn thay đổi theo thời gian cho X kênh của chương trình âm thanh đầu vào, do đó xác định N kênh gồm nội dung âm thanh được mã hóa của dòng bit được mã hóa. Trong các biến đổi này, bộ giải mã thực hiện phép nội suy trên các ma trận nguyên thủy $N \times N$ có (ví dụ, được bao gồm trong) dòng bit được mã hóa.

Theo nhóm các phương án, sáng chế đề xuất phương pháp kết xuất chương trình âm thanh đa kênh, bao gồm bước thực hiện biến đổi tuyến tính (nhân ma trận) trên các mẫu của các kênh chương trình (ví dụ, để tạo ra dạng trộn giảm của nội dung của chương trình). Sự biến đổi tuyến tính là sự biến đổi phụ thuộc vào thời gian theo nghĩa là sự biến đổi tuyến tính cần được thực hiện ở thời điểm trong chương trình (ví dụ, trên các mẫu của các kênh tương ứng với thời điểm đó) khác với sự biến đổi tuyến tính cần được thực hiện ở một thời điểm khác trong chương trình. Theo một số phương án, phương pháp này sử dụng ít nhất một ma trận hạt giống (ma trận này có thể được thực hiện dưới dạng tầng của các ma trận nguyên thủy dạng đơn vị) mà xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong chương trình (ví dụ, trên các mẫu của các kênh tương ứng với thời điểm thứ nhất), và thực hiện phép nội suy để xác định ít nhất một phiên bản nội suy của ma trận hạt giống mà xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ hai trong chương trình. Theo các phương án thông thường, phương án này được thực hiện bởi bộ giải mã (ví dụ, bộ giải mã 40 trên Fig.5 hoặc bộ giải mã 102 trên Fig.6) mà được bao gồm trong, hoặc kết hợp với, hệ thống phát lại. Thông thường, bộ giải mã được tạo cấu hình để thực hiện phục hồi không mất dữ liệu nội dung âm thanh của dòng bit âm thanh được mã hóa biểu thị chương trình, và ma trận hạt giống (và mỗi phiên bản được nội suy của ma trận hạt giống) được thực hiện dưới dạng tầng của các ma trận nguyên thủy (ví dụ, các ma trận nguyên thủy dạng đơn vị).

Thông thường, các cập nhật về ma trận kết xuất (các cập nhật của ma trận hạt giống) diễn ra không thường xuyên (ví dụ, dãy các phiên bản được cập nhật của ma trận hạt giống được bao gồm trong dòng bit âm thanh mã hóa được phân phối cho bộ giải mã, nhưng có các khoảng thời gian dài giữa các đoạn của chương trình tương ứng với các đoạn liên tiếp của các phiên bản được cập nhật), và quỹ đạo kết xuất mong muốn (ví dụ, dãy các dạng trộn mong muốn của nội dung các kênh của chương trình) giữa các lần cập nhật ma trận hạt được định rõ bằng thông số (ví dụ, bằng siêu

dữ liệu được bao gồm trong dòng bit âm thanh được mã hóa được phân phối cho bộ giải mã).

Mỗi ma trận hạt giống (của dãy các ma trận hạt giống được cập nhật) sẽ được biểu thị là $A(t_j)$, hoặc $P_k(t_j)$ nếu nó là ma trận nguyên thủy, trong đó t_j là thời gian (trong chương trình) tương ứng với ma trận hạt giống (ví dụ, thời gian tương ứng với ma trận hạt giống thứ “j”). Nếu ma trận hạt giống được thực hiện dưới dạng tầng các ma trận nguyên thủy, $P_k(t_j)$, thì chỉ số k biểu thị vị trí trong tầng của mỗi ma trận nguyên thủy này. Thông thường, ma trận thứ “k”, $P_k(t_j)$, trong tầng của ma trận nguyên thủy hoạt động trên kênh thứ “k”.

Khi sự biến đổi tuyến tính (ví dụ, đặc tả trộn giảm), $A(t)$, thay đổi nhanh, bộ mã hóa (ví dụ, bộ mã hóa thông thường) sẽ cần phải truyền các ma trận hạt giống được cập nhật thường xuyên để đạt được sự xấp xỉ gần $A(t)$.

Xem xét dãy của ma trận nguyên thủy $P_k(t1), P_k(t2), P_k(t3), \dots$, hoạt động trên cùng một kênh k nhưng ở các thời điểm khác nhau $t1, t2, t3, \dots$ chứ không phải gửi các ma trận nguyên thủy được cập nhật tại mỗi trong số các thời điểm này, phương pháp theo một phương án của sáng chế gửi, ở thời gian $t1$ (ví dụ, bao gồm trong dòng bit được mã hóa ở vị trí tương ứng với thời điểm $t1$) ma trận hạt giống nguyên thủy $P_k(t1)$, và ma trận delta hạt giống $\Delta_k(t1)$ mà xác định tỉ lệ thay đổi của các hệ số ma trận. Ví dụ, ma trận hạt giống nguyên thủy và ma trận delta hạt giống có thể có dạng:

$$P_k(t1) = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ \ddots & 1 & \ddots & \ddots & \ddots \\ \alpha_0 & \cdots & \alpha_k & \ddots & \alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Delta_k(t1) = \begin{bmatrix} 0 & 0 & \ddots & \ddots & 0 \\ \ddots & 0 & \ddots & \ddots & \ddots \\ \delta_0 & \cdots & \delta_k & \ddots & \delta_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Vì $P_k(t1)$ là ma trận nguyên thủy nên nó giống với ma trận đơn vị có kích thước $N \times N$ ngoại trừ một hàng (không tầm thường) (ví dụ, hàng này bao gồm các phần tử $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ trong ví dụ). Theo một ví dụ, ma trận $\Delta_k(t1)$ bao gồm các số 0, ngoại trừ một hàng (không bình thường) (ví dụ, hàng này bao gồm các phần tử $\delta_0, \delta_1, \dots, \delta_{N-1}$ trong ví dụ). Phần tử α_k biểu thị một trong các phần tử $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ xuất hiện trên đường chéo của $P_k(t1)$, và phần tử δ_k biểu thị một trong các phần tử $\delta_0, \delta_1, \dots, \delta_{N-1}$ xuất hiện trên đường chéo của $\Delta_k(t1)$.

Vì vậy, ma trận nguyên thủy ở thời điểm t (xuất hiện sau thời điểm t_1) được nội suy (ví dụ, bởi bước 60 hoặc 61 của bộ giải mã 42, hoặc bước 110, 111, 112, hoặc 113 của bộ giải mã 102) là:

$$P_k(t) = P_k(t_1) + f(t)\Delta_k(t_1),$$

trong đó $f(t)$ là hệ số nội suy cho thời điểm t , và $f(t_1) = 0$. Ví dụ, nếu phép nội suy tuyến tính được yêu cầu, thì hàm $f(t)$ có thể có dạng $f(t) = a^*(t - t_1)$, trong đó a là hằng số. Nếu phép nội suy được thực hiện trong bộ giải mã, thì bộ giải mã phải được tạo cấu hình để biết hàm $f(t)$. Ví dụ, siêu dữ liệu xác định hàm $f(t)$ có thể được phân phối cho bộ giải mã với dòng bit âm thanh được mã hóa cần được giải mã và kết xuất.

Trong khi phần ở trên mô tả trường hợp chung của phép nội suy các ma trận nguyên thủy, với phần tử α_k bằng 1, thì $P_k(t_1)$ là ma trận nguyên thủy dạng đơn vị mà có thể phù hợp với việc nghịch đảo không mất dữ liệu. Tuy nhiên, để duy trì tính không mất dữ liệu ở mỗi thời điểm, cũng cần phải thiết lập $\delta_k = 0$, sao cho ma trận nguyên thủy ở mỗi thời điểm này có thể phù hợp với sự nghịch đảo không mất dữ liệu.

Lưu ý rằng $P_k(t)\mathbf{x}(t) = P_k(t_1)\mathbf{x}(t) + f(t)(\Delta_k(t_1)\mathbf{x}(t))$. Vì vậy, thay vì cập nhật ma trận hạt giống nguyên thủy ở mỗi thời điểm t , có thể tính toán một cách tương đương hai tập hợp hợp kênh trung gian $P_k(t_1)\mathbf{x}(t)$ và $\Delta_k(t_1)\mathbf{x}(t)$, và kết hợp chúng với hệ số nội suy $f(t)$. Phương pháp này thường ít tốn kém hơn về mặt tính toán so với phương pháp cập nhật ma trận nguyên thủy ở mỗi thời điểm trong đó mỗi hệ số delta phải được nhân với hệ số nội suy.

Tuy nhiên, một phương pháp tương đương khác là tách $f(t)$ thành số nguyên, và phân số $f(t) - r$, và sau đó thu được áp dụng cần thiết của ma trận nguyên thủy nội suy là: $P_k(t)\mathbf{x}(t) = (P_k(t_1) + r\Delta_k(t_1))\mathbf{x}(t) + (f(t) - r)(\Delta_k(t_1)\mathbf{x}(t))$. (2)

Vì vậy, phương pháp sau này (sử dụng phương trình (2)) sẽ là sự pha trộn của hai phương pháp được mô tả trước đây.

Trong TrueHD, giá trị 0,833ms (40 mẫu ở 48 kHz) của âm thanh được định nghĩa là đơn vị truy cập. Nếu ma trận delta $\tilde{A}_k$ được định nghĩa là tỉ lệ thay đổi của ma trận nguyên thủy P_k cho mỗi đơn vị truy cập, và nếu định nghĩa $f(t) = (t - t_1) / T$,

trong đó r là độ dài của đơn vị truy cập, thì ,trong phương trình (2) tăng lên 1 cho mỗi đơn vị truy cập, và $f(t)-r$ là hàm số đơn giản của độ dịch của mẫu trong đơn vị truy cập. Vì vậy giá trị phân số $f(t)-r$ không nhất thiết phải được tính toán và có thể được suy ra một cách đơn giản từ bảng tìm kiếm được chỉ rõ bởi độ dịch trong đơn vị truy cập. Ở phần cuối của mỗi đơn vị truy cập, $P_k(t1)+r\Delta_k(t1)$ được cập nhật bằng phần bổ sung $\Delta_k(t1)$. Nói chung, r không cần tương ứng với đơn vị truy cập và thay vào đó có thể là sự phân đoạn cố định của tín hiệu, ví dụ, nó có thể là khối có độ dài 8 mẫu.

Một sự đơn giản hóa thêm nữa, cho dù chỉ là phép tính gần đúng, sẽ là bỏ qua phần phân số $f(t)-r$ hoàn toàn và cập nhật định kỳ $P_k(t1)+r\Delta_k(t1)$. Điều này về cơ bản tạo ra sự cập nhật ma trận không đổi theo từng phần, nhưng không có yêu cầu truyền các ma trận nguyên thủy thường xuyên.

Fig.3 là sơ đồ khối của hệ mạch được dùng theo một phương án của sáng chế để áp dụng ma trận nguyên thủy 4×4 (được thực hiện với thuật toán có độ chính xác hữu hạn) cho bốn kênh của chương trình âm thanh. Ma trận nguyên thủy là ma trận hạt giống nguyên thủy, một hàng không tầm thường của ma trận này bao gồm các phần tử $\alpha_0, \alpha_1, \alpha_2$, và α_3 . Dự tính rằng bốn ma trận nguyên thủy, mỗi ma trận để biến đổi các mẫu của một kênh khác trong số bốn kênh, sẽ được nối tầng để biến đổi các mẫu của tất cả bốn kênh. Hệ mạch đó có thể được sử dụng khi các ma trận nguyên thủy được cập nhật lần thứ nhất qua phép nội suy, và các ma trận nguyên thủy được cập nhật được áp dụng dựa vào dữ liệu âm thanh.

Fig.4 là sơ đồ khối của hệ mạch được dùng theo một phương án của sáng chế để áp dụng ma trận nguyên thủy 3×3 (được thực hiện với thuật toán có độ chính xác hữu hạn) cho ba kênh của chương trình âm thanh. Ma trận nguyên thủy là ma trận nguyên thủy được nội suy, được tạo ra theo phương án của sáng chế từ ma trận hạt giống nguyên thủy $P_k(t1)$, một hàng không tầm thường của ma trận này bao gồm các phần tử α_0, α_1 , và α_2 , và ma trận delta hạt giống $\Delta_k(t1)$, hàng không tầm thường của ma trận này bao gồm các phần tử δ_0, δ_1 , và δ_2 , và hàm nội suy $f(t)$. Vì vậy, ma trận nguyên thủy ở một nấc thời gian t (diễn ra sau thời điểm $t1$) được nội suy là:

$P_k(t) = P_k(t1) + f(t)\Delta_k(t1)$, trong đó $f(t)$ là hệ số nội suy cho thời điểm t (giá trị của hàm nội suy $f(t)$ tại thời điểm t), và $f(t1) = 0$. Dự tính rằng ba ma trận nguyên thủy này, mỗi ma trận dùng để biến đổi các mẫu của một trong số ba kênh khác nhau, sẽ được nối tăng để biến đổi các mẫu của tất cả ba kênh. Hệ mạch đó có thể được sử dụng khi ma trận hạt giống nguyên thủy hoặc được cập nhật một phần được áp dụng trên dữ liệu âm thanh và ma trận delta được áp dụng trên dữ liệu âm thanh và hai ma trận này được kết hợp cùng nhau bằng cách sử dụng hệ số nội suy.

Hệ mạch trên Fig.3 được tạo cấu hình để áp dụng ma trận hạt giống nguyên thủy cho bốn kênh chương trình âm thanh S1, S2, S3, và S4 (ví dụ, nhân các mẫu của các kênh với ma trận). Cụ thể là, mẫu của kênh S1 được nhân với hệ số α_0 (được nhận dạng là “m_coeff[p,0]”) của ma trận, mẫu của kênh S2 được nhân với hệ số α_1 (được nhận dạng là “m_coeff[p,1]”) của ma trận, mẫu của kênh S3 được nhân với hệ số α_2 (được nhận dạng là “m_coeff[p,2]”) của ma trận, và mẫu của kênh S4 được nhân với hệ số α_3 (được nhận dạng là “m_coeff[p,3]”) của ma trận. Các tích số được cộng vào phần tử tổng 10, và mỗi tổng xuất ra từ phần tử tổng 10 sau đó được lượng tử hóa trong bước lượng tử hóa Qss để tạo ra giá trị được lượng tử hóa mà là phiên bản được biến đổi (được bao gồm trong kênh S2’) của mẫu kênh S2. Theo một phương án thực hiện thông thường, mỗi mẫu của mỗi kênh S1, S2, S3, và S4 bao gồm 24 bit (như được biểu thị trên Fig.3), và đầu ra của mỗi phần tử nhân bao gồm 38 bit (như cũng được biểu thị trên Fig.3), và bước lượng tử hóa Qss xuất ra giá trị 24 bit được lượng tử hóa đáp lại mỗi giá trị 38 bit mà được nhập vào đó.

Hệ mạch trên Fig.4 được tạo cấu hình để áp dụng ma trận nguyên thủy được nội suy cho ba kênh chương trình âm thanh C1, C2, và C3 (ví dụ, để nhân các mẫu của các kênh với ma trận). Cụ thể là, mẫu của kênh C1 được nhân với hệ số α_0 (được nhận dạng là “m_coeff[p,0]”) của ma trận hạt giống nguyên thủy, mẫu của kênh C2 được nhân với hệ số α_1 (được nhận dạng là “m_coeff[p,1]”) của ma trận hạt giống nguyên thủy, và mẫu của kênh S3 được nhân với hệ số α_2 (được nhận dạng là “m_coeff[p,2]”) của ma trận hạt giống nguyên thủy. Các tích số được cộng vào phần tử tổng 12, và mỗi tổng xuất ra từ phần tử 12 sau đó được cộng (ở bước 14) vào giá trị tương ứng xuất ra từ bước hệ số nội suy 13. Giá trị xuất ra từ bước 14 được lượng tử hóa trong bước lượng tử hóa Qss để tạo ra giá trị được lượng tử hóa mà là phiên bản được biến đổi (bao gồm trong kênh C3’) của mẫu của kênh C3.

Mẫu kênh C1 giống nhau được nhân với hệ số δ_0 (được nhận dạng là “delta_cf[p,0]”) của ma trận delta hạt giống, mẫu của kênh C2 được nhân với hệ số δ_1 (được nhận dạng là “delta_cf[p,1]”) của ma trận delta hạt giống, và mẫu của kênh S3 được nhân với hệ số δ_2 (được nhận dạng là “delta_cf[p,2]”) của ma trận delta hạt giống. Các tích số được cộng vào phần tử tổng 11, và mỗi tổng được xuất ra từ phần tử 11 này sau đó được lượng tử hóa trong bước lượng tử hóa Qfine để tạo ra giá trị lượng tử hóa mà sau đó được nhân (ở bước phần tử nội suy 13) với giá trị hiện hành của hàm nội suy, $f(t)$.

Theo phương án thực hiện thông thường trên Fig.4, mỗi mẫu của mỗi kênh C1, C2, và C3 bao gồm 32 bit (như được biểu thị trên Fig.4), và đầu ra của mỗi trong số các phần tử tổng 11, 12, và 14 bao gồm 50 bit (như cũng được biểu thị trên Fig.4), và mỗi bước lượng tử hóa Qfine và Qss xuất ra giá trị được lượng tử hóa 32 bit để đáp lại mỗi giá trị 50 bit mà được nhập vào đó.

Ví dụ, sự thay đổi đối với hệ mạch trên Fig.4 có thể làm biến đổi vector của các mẫu gồm x kênh âm thanh, trong đó $x = 2, 4, 8$, hoặc N kênh. Dãy gồm x sự biến đổi đối với mạch trên Fig.4 có thể thực hiện phép tính nhân ma trận của x kênh này với ma trận hạt giống $x \times x$ (hoặc phiên bản nội suy của ma trận hạt giống này). Ví dụ, tầng gồm x biến thể đối với mạch trên Fig.4 có thể thực hiện các bước 60 và 47 của bộ giải mã 42 (trong đó $x = 8$), hoặc các bước 61 và 48 của bộ giải mã 42 (trong đó $x = 2$), hoặc các bước 113 và 109 của bộ giải mã 102 (trong đó $x = N$), hoặc các bước 112 và 108 của bộ giải mã 102 (trong đó $x = 8$), hoặc các bước 111 và 107 của bộ giải mã 102 (trong đó $x = 6$), hoặc các bước 110 và 106 của bộ giải mã 102 (trong đó $x = 2$).

Theo phương án trên Fig.4, ma trận hạt giống nguyên thủy và ma trận delta hạt giống được áp dụng song song cho mỗi tập hợp (vector) các mẫu đầu vào (mỗi vector này bao gồm một mẫu lấy từ mỗi trong số các kênh đầu vào).

Dựa vào Fig.6, tiếp theo sáng chế mô tả một phương án của sáng chế trong đó chương trình âm thanh cần được giải mã là chương trình âm thanh dựa trên đối tượng N kênh. Hệ thống trên Fig.6 bao gồm bộ mã hóa 100 (một phương án của bộ mã hóa theo sáng chế), hệ thống phụ phân phối 31, và bộ giải mã 102 (một phương án của bộ giải mã của sáng chế), được liên kết cùng nhau như được thể hiện. Mặc dù hệ thống phụ 102 được gọi trong bản mô tả này là “bộ giải mã”, nhưng cần phải hiểu rằng nó

có thể được thực hiện như hệ thống phát lại bao gồm hệ thống phụ giải mã (được tạo cấu hình để phân tích và giải mã dòng bit biểu thị chương trình âm thanh đa kênh được mã hóa) và các hệ thống phụ khác được tạo cấu hình để thực hiện kết xuất và ít nhất một số bước phát lại đầu ra của hệ thống phụ giải mã. Một số phương án của sáng chế là các bộ giải mã mà không được tạo cấu hình để thực hiện kết xuất và/hoặc phát lại (và sẽ thường được sử dụng với hệ thống kết xuất và/hoặc phát lại riêng). Một số phương án của sáng chế là các hệ thống phát lại (ví dụ, hệ thống phát lại bao gồm hệ thống phụ giải mã và các hệ thống phụ khác được tạo cấu hình để thực hiện kết xuất và ít nhất một số bước phát lại đầu ra của hệ thống phụ giải mã).

Trong hệ thống trên Fig.6, bộ mã hóa 100 được tạo cấu hình để mã hóa chương trình âm thanh N kênh dựa trên đối tượng dưới dạng dòng bit được mã hóa bao gồm bốn dòng phụ, và bộ giải mã 102 được tạo cấu hình để giải mã dòng bit mã hóa để kết xuất chương trình gốc N kênh (theo cách không mất dữ liệu), hoặc dạng trộn giảm 8 kênh của chương trình gốc N kênh, hoặc dạng trộn giảm 6 kênh của chương trình gốc N kênh, hoặc dạng trộn giảm 2 kênh của chương trình gốc N kênh. Bộ mã hóa 100 được ghép và được tạo cấu hình để tạo ra dòng bit mã hóa và xác nhận dòng bit mã hóa với hệ thống phân phối 31.

Hệ thống phân phối 31 được ghép và được tạo cấu hình để phân phối (ví dụ, bằng cách lưu trữ và/hoặc truyền) dòng bit mã hóa cho bộ giải mã 102. Theo một số phương án, hệ thống 31 thực hiện phân phối (ví dụ, truyền) chương trình âm thanh đa kênh được mã hóa trên hệ thống phát rộng hoặc mạng (ví dụ, internet) cho bộ giải mã 102. Theo một số phương án, hệ thống 31 lưu trữ chương trình âm thanh đa kênh được mã hóa vào vật ghi lưu trữ (ví dụ, đĩa hoặc bộ đĩa), và bộ giải mã 102 được tạo cấu hình để đọc chương trình từ vật ghi lưu trữ này.

Khối có ký hiệu “InvChAssign3” trong bộ mã hóa 100 được tạo cấu hình để thực hiện hoán vị kênh (tương đương với việc nhân với ma trận hoán vị) trên các kênh của chương trình đầu vào. Các kênh được hoán vị sau đó trải qua quá trình mã hóa ở bước 101, bước này xuất ra N kênh tín hiệu mã hóa. Các kênh tín hiệu mã hóa có thể (nhưng không cần thiết) tương ứng với các kênh loa phát lại. Các kênh tín hiệu mã hóa đôi khi được gọi là các kênh “nội bộ” vì bộ giải mã (và/hoặc hệ thống kết xuất) thường giải mã và kết xuất nội dung của các kênh tín hiệu mã hóa để phục hồi âm thanh đầu vào, sao cho các kênh tín hiệu mã hóa là “nội bộ” đối với hệ thống mã

hóa/giải mã. Việc mã hóa được thực hiện ở bước 101 là tương đương với việc nhân mỗi tập hợp mẫu của các kênh được hoán vị với ma trận mã hóa (được thực hiện dưới dạng dãy các phép nhân ma trận, được nhận dạng là $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$).

Mỗi ma trận $P_n^{-1}, \dots, P_1^{-1}$, và P_0^{-1} (và do đó tăng được áp dụng bởi bước 101) được xác định trong hệ thống phụ 103, và được cập nhật theo từng thời điểm (thường là không thường xuyên) theo ma trận trộn thay đổi theo thời gian được định rõ của N kênh của chương trình vào N kênh tín hiệu đã mã hóa được định rõ trong một khoảng thời gian.

Theo các thay đổi trong phương án làm ví dụ trên Fig.6, chương trình âm thanh đầu vào bao gồm số lượng tùy ý (N hoặc X, ở đó X lớn hơn N) các kênh. Theo các thay đổi này, N kênh chương trình âm thanh đa kênh mà được biểu thị bởi dòng bit mã hóa xuất ra từ bộ mã hóa, mà có thể được phục hồi không mất dữ liệu bởi bộ giải mã, có thể là N kênh nội dung âm thanh mà được tạo ra từ chương trình âm thanh đầu vào X kênh bằng cách thực hiện các phép toán ma trận trộn trên chương trình âm thanh đầu vào X kênh để áp dụng ma trận trộn thay đổi theo thời gian cho X kênh của chương trình âm thanh đầu vào, nhờ đó xác định nội dung âm thanh mã hóa của dòng bit mã hóa.

Hệ thống phụ xác định ma trận 103 trên Fig.6 được tạo cấu hình để tạo ra dữ liệu biểu thị các hệ số của bốn tập hợp các ma trận đầu ra (một tập hợp tương ứng với mỗi trong số bốn dòng phụ của các kênh mã hóa). Mỗi tập hợp ma trận đầu ra được cập nhật theo từng thời điểm, sao cho các hệ số cũng được cập nhật theo từng thời điểm.

Một tập hợp các ma trận đầu ra bao gồm hai ma trận kết xuất, $P_0^2(t)$, $P_1^2(t)$, mỗi trong số này là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) có kích thước 2×2 , và được dùng để kết xuất dòng phụ thứ nhất (dòng phụ trộn giảm) bao gồm hai trong số các kênh âm thanh mã hóa của dòng bit mã hóa (để kết xuất dạng trộn giảm hai kênh của âm thanh đầu vào). Tập hợp ma trận đầu ra khác có thể bao gồm nhiều đến sáu ma trận kết xuất, $P_0^6(t)$, $P_1^6(t)$, $P_2^6(t)$, $P_3^6(t)$, $P_4^6(t)$, và $P_5^6(t)$, mỗi trong số này là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) có kích thước 6×6 , và được dùng để kết xuất dòng phụ thứ hai (dòng phụ trộn giảm) bao gồm sáu kênh âm thanh được mã hóa của dòng bit mã hóa (để kết xuất dạng trộn

giảm sáu kênh của âm thanh đầu vào). Tập hợp các ma trận đầu ra khác bao gồm nhiều đến tám ma trận kết xuất, $P_0^8(t)$, $P_1^8(t)$, ..., $P_7^8(t)$, mỗi trong số này là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) có kích thước 8×8 , và được dùng để kết xuất dòng phụ thứ ba (dòng phụ trộn giảm) bao gồm tám kênh âm thanh được mã hóa của dòng bit mã hóa (để kết xuất dạng trộn giảm tám kênh của âm thanh đầu vào).

Tập hợp ma trận đầu ra khác bao gồm N ma trận kết xuất, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, mỗi trong số này là ma trận nguyên thủy (tốt hơn là ma trận nguyên thủy dạng đơn vị) có kích thước $N \times N$, và được dùng để kết xuất dòng phụ thứ tư bao gồm tất cả các kênh âm thanh mã hóa của dòng bit mã hóa (để phục hồi không mất dữ liệu chương trình âm thanh đầu vào N kênh). Đối với mỗi thời điểm, t , từng các ma trận kết xuất, $P_0^2(t)$, $P_1^2(t)$, có thể được hiểu là ma trận kết xuất cho các kênh của dòng con thứ nhất, từng các ma trận kết xuất, $P_0^6(t)$, $P_1^6(t)$, ..., $P_5^6(t)$, cũng có thể được hiểu là ma trận kết xuất cho các kênh của dòng phụ thứ hai, từng các ma trận kết xuất, $P_0^8(t)$, $P_1^8(t)$, ..., $P_7^8(t)$, cũng có thể được hiểu là ma trận kết xuất cho các kênh của dòng phụ thứ ba, và từng của các ma trận kết xuất, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, là tương đương với ma trận kết xuất cho các kênh của dòng phụ thứ tư.

Các hệ số (của mỗi ma trận kết xuất) mà được xuất ra từ hệ thống phụ 103 cho hệ thống phụ đóng gói 104 là siêu dữ liệu biểu thị độ lợi tương đối hoặc tuyệt đối của mỗi kênh cần được đưa vào trong dạng trộn của các kênh tương ứng của chương trình. Các hệ số của mỗi ma trận kết xuất (đối với một thời điểm trong chương trình) thể hiện mỗi trong số các kênh của dạng trộn nên đóng góp bao nhiêu vào việc trộn nội dung âm thanh (ở thời điểm tương ứng của dạng trộn được kết xuất) được biểu thị bởi tín hiệu nạp vào loa cho loa của hệ thống phát lại cụ thể.

N kênh âm thanh được mã hóa (xuất ra từ bước mã hóa 101), các hệ số ma trận đầu ra (được tạo ra bởi hệ thống phụ 103), và dữ liệu bổ sung (ví dụ, để bao gồm dưới dạng siêu dữ liệu trong dòng bit được mã hóa) cũng thường được xác nhận với hệ thống phụ đóng gói 104, hệ thống phụ đóng gói này tập hợp chúng thành dòng bit được mã hóa mà sau đó được xác nhận với hệ thống phân phối 31.

Dòng bit được mã hóa bao gồm dữ liệu biểu thị N kênh âm thanh được mã hóa, bốn tập hợp các ma trận đầu ra thay đổi theo thời gian (một tập hợp tương ứng

với mỗi trong số bốn dòng phụ của các kênh được mã hóa), và thường bao gồm cả dữ liệu bổ sung (ví dụ, siêu dữ liệu về nội dung âm thanh).

Bước 103 của bộ mã hóa 100 cập nhật mỗi tập hợp ma trận đầu ra (ví dụ, tập hợp P_0^2, P_1^2 , hoặc tập hợp $P_0, P_1, \dots, P_n$) theo từng thời điểm. Tập hợp các ma trận thứ nhất P_0^2, P_1^2 mà được xuất ra (ở thời điểm thứ nhất, t_1) là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy, ví dụ, các ma trận nguyên thủy dạng đơn vị) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong suốt chương trình (tức là trên các mẫu của hai kênh của đầu ra được mã hóa của bước 101, tương ứng với thời điểm thứ nhất). Tập hợp các ma trận thứ nhất $P_0^6(t), P_1^6(t), \dots, P_n^6(t)$, mà được xuất ra (ở thời điểm t_1) là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy, ví dụ, các ma trận nguyên thủy dạng đơn vị) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong suốt chương trình (tức là, trên mẫu sáu kênh của đầu ra được mã hóa của bước 101, tương ứng với thời điểm thứ nhất). Tập hợp các ma trận thứ nhất $P_0^8(t), P_1^8(t), \dots, P_n^8(t)$, mà được xuất ra (ở thời điểm t_1) là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy, ví dụ, các ma trận nguyên thủy dạng đơn vị) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong suốt chương trình (tức là trên các mẫu tám kênh của đầu ra được mã hóa của bước 101, tương ứng với thời điểm thứ nhất). Tập hợp các ma trận thứ nhất $P_0, P_1, \dots, P_n$ mà được xuất ra (ở thời điểm t_1) là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy dạng đơn vị) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm thứ nhất trong suốt chương trình (tức là, trên các mẫu của tất cả các kênh của đầu ra được mã hóa của bước 101 tương ứng với thời điểm thứ nhất).

Mỗi tập hợp ma trận được cập nhật P_0^2, P_1^2 mà được xuất ra từ bước 103 là ma trận hạt giống được cập nhật (được thực hiện dưới dạng tầng các ma trận nguyên thủy, mà cũng có thể được gọi là tầng của các ma trận hạt giống nguyên thủy) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong suốt chương trình (tức là, trên các mẫu của hai kênh của đầu ra được mã hóa của bước 101, tương ứng với thời điểm cập nhật). Mỗi tập hợp ma trận được cập nhật $P_0^6(t), P_1^6(t), \dots, P_n^6(t)$, mà được xuất ra từ tầng 103 là ma trận hạt giống được cập nhật (được thực hiện dưới dạng tầng các ma trận nguyên thủy, mà cũng có thể được gọi là

tầng của các ma trận hạt giống nguyên thủy) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong suốt chương trình (tức là, trên mẫu sáu kênh của đầu ra được mã hóa của bước 101, tương ứng với thời điểm cập nhật). Mỗi tập hợp được cập nhật của các ma trận $P_0^8(t), P_1^8(t), \dots, P_n^8(t)$, mà được xuất ra từ bước 103 là ma trận hạt giống được cập nhật (được thực hiện dưới dạng tầng các ma trận nguyên thủy, mà cũng có thể được gọi là tầng của các ma trận hạt giống nguyên thủy) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong suốt chương trình (tức là, trên mẫu của hai kênh đầu được mã hóa của bước 101, tương ứng với thời điểm cập nhật). Mỗi tập hợp ma trận đã cập nhật $P_0, P_1, \dots, P_n$ mà được xuất ra từ bước 103 cũng là ma trận hạt giống (được thực hiện dưới dạng tầng các ma trận nguyên thủy dạng đơn vị, mà cũng có thể được gọi là tầng của các ma trận hạt giống nguyên thủy dạng đơn vị) xác định sự biến đổi tuyến tính cần được thực hiện ở thời điểm cập nhật trong suốt chương trình (tức là, trên mẫu của tất cả các kênh của đầu ra được mã hóa của bước 101 tương ứng với thời điểm thứ nhất).

Bước đầu ra 103 cũng được tạo cấu hình để xuất ra các giá trị nội suy, mà (với hàm nội suy cho mỗi ma trận hạt giống) cho phép bộ giải mã 102 tạo ra các phiên bản được nội suy của các ma trận hạt giống (tương ứng với các thời điểm sau thời điểm thứ nhất, t_1 , và giữa các thời điểm cập nhật). Các giá trị nội suy (mà có thể bao gồm dữ liệu biểu thị mỗi hàm nội suy) được bao gồm bởi bước 104 trong dòng bit được mã hóa xuất ra từ bộ mã hóa 100. Các ví dụ về các giá trị nội suy này được mô tả đầu đó trong bản mô tả này (các giá trị nội suy có thể bao gồm ma trận delta cho mỗi ma trận hạt giống).

Đối với bộ giải mã 102 trên Fig.6, hệ thống phụ phân tích 105 được tạo cấu hình để chấp nhận (đọc hoặc nhận) dòng bit được mã hóa từ hệ thống phân phối 31 và để phân tích dòng bit được mã hóa. Hệ thống phụ 105 có thể hoạt động để xác nhận dòng con thứ nhất chỉ bao gồm hai kênh được mã hóa của dòng bit được mã hóa), các ma trận đầu ra ($P_0, P_1, \dots, P_n$) tương ứng với dòng phụ thứ tư (trên cùng), và các ma trận đầu ra (P_0^2, P_1^2) tương ứng với dòng con thứ nhất, với bước nhân ma trận 106 (để xử lý dòng phụ mà dẫn đến biểu diễn trộn giảm 2 kênh đối với nội dung của chương trình đầu vào N kênh gốc). Hệ thống phụ 105 có thể hoạt động để xác nhận dòng phụ thứ hai của dòng bit được mã hóa bao gồm sáu kênh đã mã hóa của dòng bit được mã hóa), và các ma trận đầu ra ($P_0^6(t), P_1^6(t), \dots, P_n^6(t)$) tương ứng với dòng phụ thứ hai,

với bước nhân ma trận 107 (để xử lý dòng phụ mà dẫn đến sự biểu diễn trộn giảm 6 kênh đối với nội dung của chương trình đầu vào N kênh gốc). Hệ thống phụ 105 có thể hoạt động để xác nhận dòng phụ thứ ba của dòng bit được mã hóa bao gồm tám kênh được mã hóa của dòng bit được mã hóa), và các ma trận đầu ra ($P_0^8(t)$, $P_1^8(t)$, ..., $P_n^8(t)$) tương ứng với dòng phụ thứ ba, với bước nhân ma trận 108 (để xử lý dòng phụ mà dẫn đến sự biểu diễn trộn giảm tám kênh đối với nội dung của chương trình đầu vào N kênh gốc). Hệ thống phụ 105 cũng có thể hoạt động để xác nhận dòng phụ thứ tư (cao nhất) của dòng bit được mã hóa (bao gồm tất cả các kênh được mã hóa của dòng bit được mã hóa), và các ma trận đầu ra tương ứng ($P_0, P_1, \dots, P_n$) với bước nhân ma trận 109 để xử lý dòng phụ nào dẫn đến sự tái tạo không mất dữ liệu đối với chương trình N kênh gốc.

Bước nội suy 113 được ghép để nhận mỗi ma trận hạt giống cho dòng phụ thứ tư (tức là, tập hợp các ma trận nguyên thủy ban đầu, $P_0, P_1, \dots, P_n$, cho thời điểm t_1 , và mỗi tập hợp các ma trận nguyên thủy được cập nhật, $P_0, P_1, \dots, P_n$) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy của mỗi ma trận hạt giống. Bước 113 được ghép và tạo cấu hình để đi qua (đến bước 109) mỗi ma trận hạt giống này, và tạo ra (và xác nhận cho bước 109) các phiên bản được nội suy của mỗi ma trận hạt giống này (mỗi phiên bản được nội suy tương ứng với thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống sau).

Bước nội suy 112 được ghép nối để nhận mỗi ma trận hạt giống cho dòng phụ thứ ba (tức là, tập hợp các ma trận nguyên thủy ban đầu, $P_0^8, P_1^8, \dots, P_n^8$, cho thời điểm t_1 , và mỗi tập hợp các ma trận nguyên thủy được cập nhật, $P_0^8, P_1^8, \dots, P_n^8$) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng được bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy cho mỗi ma trận hạt giống. Bước 112 được ghép và tạo cấu hình để đi qua (với bước 108) mỗi ma trận hạt giống này, và để tạo ra (và xác nhận cho bước 108) các phiên bản được nội suy của mỗi ma trận hạt giống (mỗi phiên bản được nội suy tương ứng với một thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống sau).

Bước nội suy 111 được ghép nối để nhận mỗi ma trận hạt giống cho dòng phụ thứ hai (tức là, tập hợp các ma trận nguyên thủy ban đầu, $P_0^6, P_1^6, \dots, P_n^6$, cho thời điểm t_1 , và mỗi tập hợp các ma trận nguyên thủy được cập nhật, $P_0^6, P_1^6, \dots, P_n^6$) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng được bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy cho mỗi ma trận hạt giống. Bước nội suy 111 được ghép và tạo cấu hình để đi qua (với bước 107) mỗi ma trận hạt giống, và tạo ra (và xác nhận cho bước 107) các phiên bản được nội suy của mỗi ma trận hạt giống (mỗi phiên bản được nội suy tương ứng với thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống sau).

Bước nội suy 110 được ghép để nhận mỗi ma trận hạt giống cho dòng con thứ nhất (tức là tập hợp các ma trận nguyên thủy ban đầu, P_0^2 và P_1^2 , cho thời điểm t_1 , và mỗi tập hợp các ma trận nguyên thủy được cập nhật, P_0^2 và P_1^2) bao gồm trong dòng bit được mã hóa, và các giá trị nội suy (cũng bao gồm trong dòng bit được mã hóa) để tạo ra các phiên bản được nội suy cho mỗi ma trận hạt giống. Bước 110 được ghép và tạo cấu hình để đi qua (với bước 106) mỗi ma trận hạt giống này, và tạo ra (và xác nhận cho bước 106) các phiên bản được nội suy của mỗi ma trận hạt giống (mỗi phiên bản được nội suy tương ứng với một thời điểm sau thời điểm thứ nhất, t_1 , và trước thời điểm cập nhật ma trận hạt giống thứ nhất, hoặc giữa các thời điểm cập nhật ma trận hạt giống sau).

Bước 106 nhân mỗi vectơ có hai mẫu âm thanh của hai kênh được mã hóa của dòng con thứ nhất với tầng được cập nhật mới nhất của các ma trận P_0^2 và P_1^2 (ví dụ, tầng của các phiên bản nội suy mới nhất của các ma trận P_0^2 và P_1^2 được tạo ra bởi bước 110), và mỗi tập hợp kết quả gồm hai mẫu được biến đổi tuyến tính trải qua phép hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối có tên "ChAssign0" để tạo ra mỗi cặp gồm các mẫu của dạng trộn giảm 2 kênh yêu cầu của N kênh âm thanh gốc. Một loạt các phép toán lập ma trận được thực hiện trong bộ mã hóa 40 và bộ giải mã 102 tương đương với việc áp dụng đặc tả ma trận trộn giảm để biến đổi N kênh âm thanh đầu vào thành dạng trộn giảm 2 kênh.

Bước 107 nhân mỗi vectơ gồm sáu mẫu âm thanh của sáu kênh được mã hóa của dòng phụ thứ hai với tầng được cập nhật mới nhất của các ma trận $P_0^6, \dots, P_n^6$ (ví dụ tầng của các phiên bản mới nhất được nội suy của các ma trận $P_0^6, \dots, P_n^6$ được

tạo ra bởi tầng 111), và mỗi tập hợp kết quả của sáu mẫu được biến đổi tuyến tính trải qua sự hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign1” để thu được mỗi tập hợp mẫu của dạng trộn giảm 6 kênh yêu cầu của N kênh âm thanh gốc. Tầng phép toán lập ma trận được thực hiện trong bộ mã hóa 100 và bộ giải mã 102 là tương đương với việc áp dụng đặc tả ma trận trộn giảm mà biến đổi N kênh âm thanh đầu vào thành dạng trộn giảm 6 kênh.

Bước 108 nhân mỗi vectơ của tám mẫu âm thanh của tám kênh được mã hóa (của dòng phụ thứ ba nhân với tầng được cập nhật mới nhất của các ma trận $P_0^8, \dots, P_n^8$ (ví dụ, tầng của các phiên bản mới nhất được nội suy của các ma trận $P_0^8, \dots, P_n^8$ được tạo ra bởi bước 112), và mỗi tập hợp thu được của tám mẫu được biến đổi tuyến tính trải qua sự hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign2” để tạo ra mỗi cặp mẫu của ma dạng trộn giảm tám kênh yêu cầu của N kênh âm thanh gốc. Tầng các phép toán lập ma trận được thực hiện trong bộ mã hóa 100 và bộ giải mã 102 là tương đương với việc áp dụng đặc tả ma trận trộn giảm mà biến đổi N kênh âm thanh đầu vào thành dạng trộn giảm 8 kênh.

Bước 109 nhân mỗi vectơ của N mẫu âm thanh (một mẫu từ mỗi trong số N kênh được mã hóa đầu đủ của dòng bit được mã hóa) với tầng được cập nhật mới nhất của các ma trận $P_0, P_1, \dots, P_n$ (ví dụ, tầng của các phiên bản mới nhất được nội suy của các ma trận $P_0, P_1, \dots, P_n$ được tạo ra bởi bước 113) và mỗi tập hợp kết quả của N các mẫu được biến đổi tuyến tính trải qua sự hoán vị kênh (tương đương với phép nhân với ma trận hoán vị) được biểu diễn bởi khối có tên “ChAssign3” để thu được mỗi tập hợp N mẫu của chương trình N kênh gốc được phục hồi không mất dữ liệu. Để cho âm thanh N kênh đầu ra là giống chính xác như the âm thanh N kênh đầu vào (để đạt được đặc điểm “không mất dữ liệu” của hệ thống), các phép toán lập ma trận được thực hiện trong bộ mã hóa 100 phải chính xác là (bao gồm các tác dụng lượng tử hóa) nghịch đảo của các phép toán lập ma trận được thực hiện trong bộ giải mã 102 trên dòng phụ thứ tư của dòng bit được mã hóa (ví dụ, mỗi lần nhân ở bước 109 của bộ giải mã 102 với tầng các ma trận $P_0, P_1, \dots, P_n$). Vì vậy, trên Fig.6, các phép toán lập ma trận ở bước 103 của bộ mã hóa 100 được xác định dưới dạng tầng các ma trận nghịch

đảo của các ma trận $P_0, P_1, \dots, P_n$, trong dãy đối diện được áp dụng ở bước 109 của bộ giải mã 102, tức là: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

Theo một số phương án thực hiện, hệ thống phụ phân tích 105 được tạo cấu hình để trích từ kiểm tra từ dòng bit được mã hóa, và bước 109 được tạo cấu hình để xác minh xem N kênh (của ít nhất một đoạn của chương trình âm thanh đa kênh) được phục hồi bởi bước 109 có được phục hồi đúng không, bằng cách so sánh từ kiểm tra thứ hai thu được (ví dụ, bởi bước 109) từ các mẫu âm thanh được tạo ra bởi bước 109 với từ kiểm tra được trích từ dòng bit được mã hóa.

Bước “ChAssign3” của bộ giải mã 102 áp dụng cho đầu ra của bước 109 sự nghịch đảo của hoán vị kênh được áp dụng bởi bộ mã hóa 100 (ví dụ, ma trận hoán vị được thể hiện bởi bước “ChAssign3” của bộ giải mã 102 là nghịch đảo của ma trận được biểu diễn bởi phân tử “InvChAssign3” của bộ mã hóa 100).

Trong các biến thể trên các hệ thống phụ 100 và 102 của hệ thống được thể hiện trên Fig.6, một hoặc nhiều phần tử được lược bỏ hoặc đơn vị xử lý dữ liệu âm thanh bổ sung đưa vào.

Các hệ số ma trận kết xuất $P_0^8, \dots, P_n^8$ (hoặc $P_0^6, \dots, P_n^6$, hoặc P_0^2 và P_1^2) được đưa vào bước 108 (hoặc 107 hoặc 106) của bộ giải mã 100 là siêu dữ liệu (ví dụ, siêu dữ liệu về vị trí không gian) của dòng bit được mã hóa, các siêu dữ liệu này biểu thị (hoặc có thể được xử lý cùng với dữ liệu khác cần biểu thị) độ lợi tương đối hoặc tuyệt đối của mỗi kênh loa cần được bao gồm trong dạng trộn giảm của các kênh của nội dung N kênh gốc được mã hóa bởi bộ mã hóa 100.

Trái lại, cấu hình của hệ thống loa phát lại cần sử dụng để kết xuất tập hợp đầy đủ của các kênh của chương trình âm thanh dựa trên đối tượng (mà được phục hồi không mất dữ liệu bởi bộ giải mã 102) thông thường không được biết tại thời điểm dòng bit đã mã hóa được tạo ra bởi bộ mã hóa 100. N kênh được phục hồi không mất dữ liệu bởi bộ giải mã 102 có thể cần được xử lý (ví dụ, trong hệ thống kết xuất nằm ở trong bộ giải mã 102 (nhưng không được thể hiện trên Fig.6) hoặc được liên kết với bộ giải mã 102) bằng dữ liệu khác (ví dụ, dữ liệu biểu thị cấu hình của hệ thống loa phát lại cụ thể) để xác định mỗi kênh của chương trình phải đóng góp bao nhiêu vào dạng trộn của nội dung âm thanh (ở mỗi thời điểm của dạng trộn được kết xuất) được biểu thị bởi tín hiệu nạp cho vào loa cho loa của hệ thống phát lại cụ thể. Hệ thống

kết xuất như vậy có thể xử lý siêu dữ liệu về quỹ đạo không gian ở (hoặc gắn với) mỗi đối tượng phục hồi không mất dữ liệu, để xác định các tín hiệu nạp cho vào loa cho các loa của hệ thống loa phát lại cụ thể cần sử dụng để phát lại nội dung phục hồi không mất dữ liệu.

Theo một số phương án của bộ giải mã theo sáng chế, bộ mã hóa được cung cấp (hoặc tạo ra) đặc tả thay đổi động $A^{(t)}$ biểu thị cách thức biến đổi tất cả các kênh của chương trình âm thanh N kênh (ví dụ, chương trình âm thanh dựa trên đối tượng) thành tập hợp N kênh được mã hóa, và ít nhất một đặc tả thay đổi động của dạng trộn giảm biểu thị mỗi dạng trộn giảm của nội dung của N kênh được mã hóa thành biểu diễn M1 kênh (trong đó M1 nhỏ hơn N, ví dụ, M1 = 2, hoặc M1 = 8, nếu N lớn hơn 8). Theo một số phương án, nhiệm vụ của bộ mã hóa là đóng gói âm thanh và dữ liệu được mã hóa biểu thị mỗi đặc tả thay đổi động như vậy thành dòng bit được mã hóa có định dạng đã định trước (ví dụ, dòng bit TrueHD). Ví dụ, điều này có thể được thực hiện sao cho bộ giải mã kế thừa (ví dụ, bộ giải mã TrueHD kế thừa) có khả năng phục hồi ít nhất một phần biểu diễn trộn giảm (có M1 kênh), trong khi đó bộ giải mã được nâng cấp có thể được sử dụng để phục hồi (không mất dữ liệu) chương trình âm thanh N kênh gốc. Căn cứ vào đặc tả thay đổi động, bộ mã hóa có thể giả định rằng bộ giải mã sẽ xác định các ma trận nguyên thủy được nội suy $P_0, P_1, \dots, P_n$ từ các giá trị nội suy (ví dụ, các thông tin về ma trận hạt giống nguyên thủy và ma trận delta hạt giống) được bao gồm trong dòng bit được mã hóa cần được phân phối cho bộ giải mã. Sau đó, bộ giải mã thực hiện phép nội suy để xác định các ma trận nguyên thủy được nội suy để đảo ngược các phép toán của bộ mã hóa mà đã tạo ra nội dung âm thanh được mã hóa của dòng bit được mã hóa (ví dụ, để phục hồi không mất dữ liệu nội dung mà được mã hóa, bằng cách thực hiện các phép toán ma trận, trong bộ mã hóa). Tùy ý, bộ mã hóa có thể chọn các ma trận nguyên thủy cho các dòng phụ thấp hơn (ví dụ, các dòng phụ biểu thị các dạng trộn giảm của nội dung của dòng phụ N kênh phía trên) là các ma trận nguyên thủy không được nội suy (và bao gồm dãy của tập hợp các ma trận nguyên thủy không được nội suy như vậy trong dòng bit được mã hóa), trong khi đó cũng giả định rằng bộ giải mã sẽ xác định các ma trận nguyên thủy được nội suy ($P_0, P_1, \dots, P_n$) để phục hồi không mất dữ liệu đối với nội dung của dòng phụ (N kênh) phía trên từ các giá trị nội suy (ví dụ, các thông tin về ma trận hạt giống

nguyên thủy và ma trận delta hạt giống) được bao gồm trong dòng bit được mã hóa cần được phân phối cho bộ giải mã.

Ví dụ, bộ mã hóa (ví dụ, bước 44 của bộ mã hóa 40, hoặc bước 103 của bộ mã hóa 100) có thể được tạo cấu hình để chọn các ma trận hạt giống nguyên thủy và ma trận delta hạt giống (để sử dụng với hàm nội suy, $f(t)$), bằng cách lấy mẫu đặc tả $A(t)$ ở các thời điểm thời gian khác nhau $t_1, t_2, t_3, \dots$ (mà có thể được đặt gần nhau), tạo ra các ma trận hạt giống nguyên thủy tương ứng (ví dụ, như trong bộ mã hóa TrueHD thông thường) và sau đó để tính toán tỉ lệ thay đổi của các phần tử riêng lẻ ở các ma trận hạt giống nguyên thủy để tính toán các giá trị nội suy (ví dụ, thông tin “delta” biểu thị dãy các ma trận delta hạt giống). Tập hợp các ma trận thứ nhất nguyên thủy hạt sẽ là các ma trận nguyên thủy được tạo ra từ đặc tả cho thời điểm thứ nhất trong số các thời điểm thời gian như vậy, $A(t_1)$. Có thể là tập hợp con của các ma trận nguyên thủy không thể thay đổi chút nào theo thời gian, trong trường hợp đó bộ giải mã sẽ phản hồi các thông tin kiểm soát phù hợp trong dòng bit được mã hóa bằng cách tính toán dứt điểm thông tin delta tương ứng bất kỳ (ví dụ, để thiết lập tỉ lệ thay đổi của tập hợp con như vậy của các ma trận nguyên thủy là 0).

Các thay đổi đối với phương án trên Fig.6 của bộ mã hóa và bộ giải mã theo theo sáng chế có thể loại bỏ sự nội suy cho một số (ví dụ, ít nhất một) trong số các dòng phụ của dòng bit được mã hóa. Ví dụ, các bước nội suy 110, 111, và 112 có thể được loại bỏ, và các ma trận tương ứng P_0^2, P_1^2 , và $P_0^6, P_1^6, \dots, P_n^6$, và $P_0^8, P_1^8, \dots, P_n^8$, có thể được cập nhật (trong dòng bit được mã hóa) với tần số đầy đủ sao cho phép nội suy giữa các thời điểm mà tại các thời điểm đó chúng được cập nhật là không cần thiết. Một ví dụ, khác, nếu các ma trận $P_0^6, P_1^6, \dots, P_n^6$, được cập nhật với tần số đầy đủ sao cho phép nội suy ở các thời điểm giữa các lần cập nhật là không cần thiết, thì bước nội suy 111 là không cần thiết và có thể được lược bỏ. Vì vậy, bộ giải mã thông thường (không được tạo cấu hình theo sáng chế để thực hiện phép nội suy) có thể kết xuất biểu diễn trộn giảm 6 kênh để đáp lại dòng bit được mã hóa.

Như nêu trên đây, các đặc tả của ma trận kết xuất động (ví dụ, $A(t)$) có thể bắt nguồn không chỉ từ nhu cầu kết xuất chương trình âm thanh dựa trên đối tượng, mà còn do nhu cầu thực hiện sự bảo vệ kẹp. Các ma trận nguyên thủy được nội suy có thể thực hiện sự biến đổi nhanh hơn đối với và thoát ra khỏi sự bảo vệ kẹp của dạng trộn giảm, cũng như giảm tốc độ dữ liệu cần thiết để truyền các hệ số lập ma trận.

Các tác giả sáng chế tiếp theo mô tả ví dụ, về việc hoạt động phương án thực hiện của hệ thống trên Fig.6. Trong trường hợp này, chương trình đầu vào N kênh là chương trình âm thanh dựa trên đối tượng ba kênh bao gồm kênh đệm, C, và hai kênh đối tượng, U và V. Mong muốn là chương trình cần được mã hóa để vận chuyển thông qua dòng TrueHD có hai dòng phụ sao cho dạng trộn giảm 2 kênh (dạng kết xuất chương trình vào thiết lập loa hai kênh) có thể được tìm kiếm bằng cách sử dụng dòng con thứ nhất và chương trình đầu vào 3 kênh gốc có thể được phục hồi không mất dữ liệu bằng cách sử dụng cả hai dòng phụ.

Giả sử phương trình kết xuất (hoặc phương trình trộn giảm) từ chương trình đầu vào thành dạng trộn 2 kênh được cho là:

$$A_2(t) = \begin{bmatrix} 0,707 & \sin(\nu t) & \cos(\nu t) \\ 0,707 & \cos(\nu t) & \sin(\nu t) \end{bmatrix}$$

trong đó cột thứ nhất tương ứng với các độ lợi của kênh đệm (kênh trung tâm, C) mà nạp bằng nhau vào các kênh L và R. Các cột thứ hai và thứ ba, lần lượt, tương ứng với kênh đối tượng U và kênh đối tượng V. Hàng thứ nhất tương ứng với L kênh của dạng trộn 2 kênh và hàng thứ hai tương ứng với R kênh. Hai đối tượng đang chuyển động hướng về nhau với tốc độ được xác định bởi .

Các tác giả sáng chế sẽ kiểm tra các ma trận kết xuất ở ba thời điểm thời gian khác nhau t_1 , t_2 và t_3 . Trong ví dụ, này, các tác giả sáng chế sẽ giả định $t_1 = 0$, tức là $A_2(t_1) = \begin{bmatrix} 0,707 & 0 & 1 \\ 0,707 & 1 & 0 \end{bmatrix}$. Nói cách khác, ở thời điểm t_1 , đối tượng U hoàn toàn nạp vào R và đối tượng V hoàn toàn trộn giảm xuống L. Vì các đối tượng chuyển động hướng về nhau, nên sự đóng góp của chúng cho loa khác tăng lên. Để phát triển ví dụ, hơn nữa, cho phép tác giả sáng chế nói rằng $\nu = \frac{\pi}{4} \times \frac{1}{40T}$, ở đó T là độ dài của đơn vị truy cập (thông thường là 0,8333 ms hoặc 40 mẫu ở tốc độ lấy mẫu 48kHz). Vì vậy, với thời gian $t = 40T$ hai đối tượng là ở trung tâm của cảnh. Các tác giả sáng chế bây giờ sẽ cân nhắc $t_2 = 15T$ và $t_3 = 30T$, sao cho:

$$A_2(t_2) = \begin{bmatrix} 0,707 & 0,2903 & 0,9569 \\ 0,707 & 0,9569 & 0,2902 \end{bmatrix}$$

$$A_2(t_3) = \begin{bmatrix} 0,707 & 0,5556 & 0,8315 \\ 0,707 & 0,8315 & 0,5556 \end{bmatrix}$$

Tác giả sáng chế cân nhắc phân tích đặc tả được cung cấp $A_2(t)$ thành các ma trận nguyên thủy đầu vào và đầu ra. Để đơn giản, tác giả sáng chế xin phép giả định rằng các ma trận P_0^2, P_1^2 là các ma trận đơn vị và chAssign0 (trong bộ giải mã 102) là ký hiệu gán kênh đơn vị, ví dụ, bằng với sự hoán vị bình thường (ma trận đơn vị).

Các tác giả sáng chế có thể thấy rằng:

$$\begin{bmatrix} 0,707 & 0 & 1 \\ 0,707 & 1 & 0 \\ 1 & -1,414 & 4,243 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & -0,707 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 0,707 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1,414 & 4,243 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t1) \quad P_1^{-1}(t1) \quad P_2^{-1}(t1) \quad \text{InvChAssign1}(t1)$

Hai hàng thứ nhất của tích số nêu trên chính xác là đặc tả $A_2(t1)$. Nói cách khác, các ma trận nguyên thủy $P_0^{-1}(t1)$, $P_1^{-1}(t1)$, $P_2^{-1}(t1)$, và sự gán kênh được biểu thị bởi InvChAssign1(t1) cùng nhau làm biến đổi kênh đầu vào C, Đối tượng U, và Đối tượng V thành ba kênh nội bộ, hai kênh đầu trong số các kênh này chính xác là các dạng trộn giảm yêu cầu L và R. Vì vậy, sự phân tích trên đây của $A(t1)$ thành các ma trận nguyên thủy $P_0^{-1}(t1)$, $P_1^{-1}(t1)$, $P_2^{-1}(t1)$, và gán kênh InvChAssign1(t1) là lựa chọn hợp lệ của các ma trận nguyên thủy đầu vào nếu các ma trận nguyên thủy đầu ra và sự gán kênh cho sự trình diễn hai kênh được chọn là các ma trận đơn vị. Lưu ý rằng các ma trận nguyên thủy đầu vào có thể đảo ngược không mất dữ liệu để phục hồi C, Đối tượng U và Đối tượng V bởi bộ giải mã mà hoạt động trên tất cả ba kênh nội bộ. Tuy nhiên, bộ giải mã hai kênh sẽ chỉ cần kênh nội bộ 1 và 2 và áp dụng các ma trận nguyên thủy đầu ra P_0^2, P_1^2 và chAssign0, mà trong trường hợp này đều là đồng nhất thức.

Tương tự các tác giả sáng chế có thể xác định:

$$\begin{bmatrix} 0,707 & 0,2903 & 0,9569 \\ 0,707 & 0,9569 & 0,2903 \\ 1 & -1,004 & 4,890 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1,666 & 1 & -0,4713 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2,5 & 0,707 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1,003 & 4,889 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t2) \quad P_1^{-1}(t2) \quad P_2^{-1}(t2) \quad \text{InvChAssign1}(t2)$

trong đó hai hàng thứ nhất là giống với $A(t2)$, và

$$\begin{bmatrix} 0,707 & 0,5556 & 0,8315 \\ 0,707 & 0,8315 & 0,5556 \\ 1 & -0,628 & 7,717 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1,2759 & 1 & -0,1950 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -4,624 & 0,707 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0,628 & 7,717 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t3) \qquad P_1^{-1}(t3) \qquad P_2^{-1}(t3) \qquad \text{InvChAssign1}(t3)$

trong đó hai hàng thứ nhất là giống với $A(t3)$.

Bộ mã hóa TrueHD kế thừa (mà không thực hiện sáng chế) có thể chọn để truyền (sự đảo ngược của) các ma trận nguyên thủy được thiết kế trên đây với thời gian $t1$, $t2$, và $t3$, tức là, $\{P_0(t1), P_1(t1), P_2(t1)\}, \{P_0(t2), P_1(t2), P_2(t2)\}, \{P_0(t3), P_1(t3), P_2(t3)\}$. Trong trường hợp này đặc tả với thời điểm bất kỳ t nằm giữa $t1$ và $t2$ được lấy gần đúng bởi đặc tả với thời gian $A(t1)$, và nằm giữa $t2$ và $t3$ được lấy gần đúng bởi $A(t2)$.

Trong phương án ví dụ, của hệ thống trên Fig.6, ma trận nguyên thủy $P_0^{-1}(t)$ ở $t = t1$, hoặc $t = t2$, hoặc $t = t3$ hoạt động trên cùng một kênh (kênh 2), tức là, hàng không tầm thường trong cả ba trường hợp này là hàng thứ hai. Tương tự là trường hợp với $P_1^{-1}(t)$ và $P_2^{-1}(t)$. Ngoài ra, InvChAssign1 ở mỗi thời điểm thời gian là giống nhau.

Vì vậy, để thực hiện việc mã hóa bằng phương án ví dụ, của bộ mã hóa 100 trên Fig.6), các tác giả sáng chế có thể tính toán các ma trận delta:

$$\Delta_0(t1) = \frac{P_0(t2) - P_0(t1)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0,0222 & 0 & -0,0157 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_1(t1) = \frac{P_1(t2) - P_1(t1)}{15} = \begin{bmatrix} 0 & 0,0333 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_2(t1) = \frac{P_2(t2) - P_2(t1)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0,0274 & -0,0431 & 0 \end{bmatrix}$$

và

$$\Delta_0(t2) = \frac{P_0(t3) - P_0(t2)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0,0261 & 0 & -0,0184 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_1(t2) = \frac{P_1(t3) - P_1(t2)}{15} = \begin{bmatrix} 0 & 0,1416 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_2(t2) = \frac{P_2(t3) - P_2(t2)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0,0250 & -0,01885 & 0 \end{bmatrix}$$

Khác với bộ mã hóa TrueHD kế thừa, bộ mã hóa TrueHD được phép lập ma trận có nội suy (phương án ví dụ, của bộ mã hóa 100 trên Fig.6) có thể chọn gửi các ma trận hạt giống (nguyên thủy và đenta)

$$\{P_0(t1), P_1(t1), P_2(t1)\}, \{\Delta_0(t1), \Delta_1(t1), \Delta_2(t1)\}, \{\Delta_0(t2), \Delta_1(t2), \Delta_2(t2)\}.$$

Các ma trận nguyên thủy và các ma trận đenta ở thời điểm thời gian trung gian bất kỳ được tạo ra bởi phép nội suy. Các phương trình trộn trộn giảm đạt được ở thời điểm cho trước t ở giữa $t1$ và $t2$ có thể được tạo ra dưới dạng hai hàng thứ nhất của tích số:

$$\left(P_0^{-1}(t1) - \Delta_0(t1) * \frac{t}{T} \right) \left(P_1^{-1}(t1) - \Delta_1(t1) * \frac{t}{T} \right) \left(P_2^{-1}(t1) - \Delta_2(t1) * \frac{t}{T} \right) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \text{ và}$$

giữa $t2$ và $t3$, dưới dạng

$$\left(P_0^{-1}(t2) - \Delta_0(t2) * \frac{t}{T} \right) \left(P_1^{-1}(t2) - \Delta_1(t2) * \frac{t}{T} \right) \left(P_2^{-1}(t2) - \Delta_2(t2) * \frac{t}{T} \right) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Trong các ma trận trên đây $\{P_0(t2), P_1(t2), P_2(t2)\}$ thực tế không được truyền nhưng được tạo ra dưới dạng các ma trận nguyên thủy của điểm cuối cùng của phép nội suy với các ma trận đenta $\{\Delta_0(t1), \Delta_1(t1), \Delta_2(t1)\}$.

Vì vậy, các tác giả sáng chế biết các phương trình trộn trộn giảm đạt được ở mỗi thời điểm “ t ” cho cả hai trường hợp trên đây. Các tác giả sáng chế có thể vì vậy tính toán sự không phù hợp giữa phép tính xấp xỉ với thời gian cho trước “ t ” và đặc tả thật trong thời điểm thời gian đó. Fig.7 là đồ thị về tổng các sai số bình phương giữa đặc tả đạt được và đặc tả thật ở các thời điểm thời gian khác nhau t , nhờ sử dụng phép nội

suy của các ma trận nguyên thủy (đường cong được ký hiệu là “Lập ma trận được nội suy”) và với các ma trận nguyên thủy không đổi theo từng phần (không được nội suy) (đường cong được ký hiệu là “Lập ma trận không được nội suy”). Qua Fig.7, rõ ràng là lập ma trận được nội suy dẫn đến đạt được đặc tả $A_2(t)$ một cách gần đúng hơn đáng kể so với việc lập ma trận không được nội suy trong vùng từ 0 đến 600s ($t_1 - t_2$). Để đạt được mức độ méo giống nhau bằng việc lập ma trận không được nội suy, chuyên gia có thể phải gửi các cập nhật ma trận ở nhiều điểm ở giữa t_1 và t_2 .

Việc lập ma trận không được nội suy có thể dẫn đến đạt được dạng trộn giảm mà gần hơn với đặc tả thật ở một số thời điểm thời gian tức thì (ví dụ, nằm trong khoảng từ 600 giây đến 900 giây trong ví dụ, trên Fig.7) nhưng sai số trong việc lập ma trận không được nội suy liên tục tích tụ theo thời gian giảm dần đến lần cập nhật ma trận tiếp theo trong khi đó sai số với việc lập ma trận nội suy giảm bớt gần với các điểm cập nhật (trong trường hợp này ở $t_3 = 30 \cdot T = 1200$ giây). Sau số trong việc lập ma trận nội suy có thể còn được giảm nữa bằng cách gửi thông tin cập nhật delta khác ở giữa t_2 và t_3 .

Các phương án khác nhau của sáng chế thực hiện một hoặc nhiều trong số các dấu hiệu dưới đây:

1. sự biến đổi của một tập hợp các kênh âm thanh thành số lượng bằng nhau của các kênh âm thanh khác bằng cách áp dụng dãy các ma trận nguyên thủy (tốt hơn là, các ma trận nguyên thủy dạng đơn vị) ở đó mỗi trong ít nhất một số ma trận nguyên thủy là ma trận nguyên thủy được nội suy được tính toán dưới dạng tổ hợp tuyến tính (được xác định theo hàm nội suy) của ma trận hạt giống nguyên thủy và ma trận delta hạt giống hoạt động trên cùng một kênh âm thanh. Hệ số của tổ hợp tuyến tính được xác định bởi nhóm nội suy (ví dụ, mỗi hệ số của ma trận nguyên thủy được nội suy là tổ hợp tuyến tính $A + f(t)B$, ở đó A là hệ số của ma trận hạt giống nguyên thủy, B là hệ số tương ứng của ma trận delta hạt giống, và $f(t)$ là giá trị của hàm nội suy ở thời gian, t , gắn với ma trận nguyên thủy được nội suy). Trong một số trường hợp, sự biến đổi được thực hiện trên nội dung âm thanh được mã hóa của dòng bit được mã hóa để thực hiện sự phục hồi không mất dữ liệu đối với nội dung âm thanh được mã hóa để tạo ra dòng bit được mã hóa;

2. sự biến đổi theo dấu hiệu 1 trên đây, trong đó việc áp dụng ma trận nguyên thủy được nội suy đạt được bằng cách áp dụng ma trận hạt giống nguyên thủy và ma

trận đenta hạt giống một cách riêng rẽ trên các kênh âm thanh cần biến đổi, và kết hợp tuyến tính các mẫu âm thanh thu được (ví dụ, các phép nhân ma trận của ma trận hạt giống nguyên thủy được thực hiện song song với các phép nhân ma trận của ma trận đenta hạt giống, như trong mạch trên Fig.4);

3. sự biến đổi theo dấu hiệu 1 trên đây, trong đó phần tử nội suy được duy trì hầu như không đổi trong một số khoảng (ví dụ, các khoảng ngắn) của các mẫu dòng bit được mã hóa, và ma trận hạt giống nguyên thủy mới gần nhất được cập nhật (bằng phép nội suy) chỉ trong các khoảng trong đó phần tử nội suy thay đổi (ví dụ, để giảm bớt độ phức tạp xử lý trong bộ giải mã);

4. sự biến đổi theo dấu hiệu 1 trên đây, trong đó các ma trận nguyên thủy được nội suy là các ma trận nguyên thủy dạng đơn vị. Trong trường hợp này, phép nhân của tầng các ma trận nguyên thủy dạng đơn vị (trong bộ mã hóa) tiếp đó là phép nhân (trong bộ giải mã) của tầng các đảo ngược của chúng có thể được thực hiện không mất dữ liệu với việc xử lý có độ chính xác hữu hạn;

5. sự biến đổi theo dấu hiệu 1 trên đây, trong đó sự biến đổi được thực hiện trong bộ giải mã âm thanh để trích các kênh âm thanh và ma trận hạt giống được mã hóa từ dòng bit được mã hóa, trong đó bộ giải mã tốt hơn là được tạo cấu hình để xác minh xem âm thanh được giải mã (sau khi được lập ma trận) có được xác định đúng không, bằng cách so sánh từ kiểm tra có được từ âm thanh sau khi được lập ma trận với từ kiểm tra được trích từ dòng bit được mã hóa;

6. sự biến đổi theo dấu hiệu 1 trên đây, trong đó sự biến đổi được thực hiện trong bộ giải mã của hệ thống mã hóa âm thanh không mất dữ liệu mà trích các kênh âm thanh và ma trận hạt giống được mã hóa từ dòng bit được mã hóa, và các kênh âm thanh được mã hóa đã được tạo ra bởi bộ mã hóa tương ứng mà áp dụng các ma trận nguyên thủy nghịch đảo không mất dữ liệu cho âm thanh đầu vào, nhờ đó mã hóa âm thanh đầu vào theo cách không mất dữ liệu vào dòng bit;

7. sự biến đổi theo dấu hiệu 1 trên đây, trong đó sự biến đổi được thực hiện trong bộ giải mã mà nhân các kênh được mã hóa nhận được với tầng các ma trận nguyên thủy, và chỉ tập hợp con của các ma trận nguyên thủy được xác định bởi phép nội suy (ví dụ, các phiên bản được cập nhật của các ma trận nguyên thủy còn lại có

thể được phân phối cho bộ giải mã theo từng thời điểm, nhưng bộ giải mã không thực hiện phép nội suy để cập nhật chúng);

8. sự biến đổi theo dấu hiệu 1 trên đây, trong đó các ma trận hạt giống nguyên thủy, các ma trận delta hạt giống, và hàm nội suy được chọn sao cho tập hợp con của các kênh được mã hóa do bộ mã hóa tạo ra có thể được biến đổi thông qua các phép toán lập ma trận được thực hiện (sử dụng các ma trận và hàm nội suy) bởi bộ giải mã để đạt được các dạng trộn giảm cụ thể của âm thanh gốc được mã hóa bởi bộ mã hóa;

9. sự biến đổi theo dấu hiệu trên đây 8, ở đó âm thanh gốc là chương trình âm thanh dựa trên đối tượng, và các dạng trộn giảm cụ thể tương ứng với việc kết xuất các kênh của chương trình thành các bố cục của loa tĩnh (ví dụ, âm lập thể, hoặc kênh 5.1, hoặc kênh 7.1);

10. sự biến đổi theo dấu hiệu 9 trên đây, ở đó các đối tượng âm thanh được biểu thị bởi chương trình là động sao cho các đặc tả trộn giảm đối với bố cục cụ thể của loa tĩnh thay đổi ngay lập tức, với sự thay đổi ngay lập tức được điều tiết bằng cách thực hiện việc lập ma trận nội suy trên các kênh âm thanh được mã hóa để tạo ra biểu diễn trộn giảm;

11. sự biến đổi theo dấu hiệu 1 trên đây, trong đó bộ giải mã được phép nội suy (được tạo cấu hình để thực hiện phép nội suy theo phương án của sáng chế) cũng có khả năng giải mã các dòng phụ của dòng bit được mã hóa theo cú pháp kế thừa mà không thực hiện phép nội suy để xác định ma trận được nội suy;

12. sự biến đổi theo dấu hiệu 1 trên đây, ở đó các ma trận nguyên thủy được thiết kế để khai thác sự tương quan giữa các kênh để đạt được độ nén tốt hơn; và

13. sự biến đổi theo dấu hiệu 1 trên đây, trong đó lập ma trận nội suy được sử dụng để đạt được đặc tả trộn giảm động được thiết kế để bảo vệ kẹp.

Giả sử rằng các ma trận trộn giảm được tạo ra nhờ sử dụng phép nội suy theo phương án của sáng chế (để phục hồi các biểu diễn trộn giảm từ dòng bit được mã hóa) thông thường thay đổi liên tục khi tín hiệu nguồn là chương trình âm thanh dựa trên đối tượng, các ma trận hạt giống nguyên thủy được sử dụng (ví dụ, được bao gồm trong dòng bit được mã hóa) theo các phương án thông thường của sáng chế thông thường cần được cập nhật để phục hồi các biểu diễn trộn giảm đó.

Nếu các ma trận hạt giống nguyên thủy được cập nhật thường xuyên, để tính gần xấp xỉ cho đặc tả ma trận thay đổi liên tục, dòng bit được mã hóa thường bao gồm dữ liệu biểu thị dãy các tầng của các tập hợp ma trận hạt giống nguyên thủy, $\{P_0(t1), P_1(t1), \dots, P_n(t1)\}$, $\{P_0(t2), P_1(t2), \dots, P_n(t2)\}$, $\{P_0(t3), P_1(t3), \dots, P_n(t3)\}$, và v.v. Điều này cho phép bộ giải mã phục hồi tầng được định rõ trong số các ma trận ở mỗi thời điểm thời gian cập nhật $t1, t2, t3, \dots$. Vì các ma trận kết xuất được biểu thị trong các hệ thống kết xuất chương trình âm thanh dựa trên đối tượng thường thay đổi liên tục kịp thời, nên mỗi ma trận hạt giống nguyên thủy (trong dãy tầng các ma trận hạt giống nguyên thủy được bao gồm trong dòng bit được mã hóa) có thể có cấu hình ma trận nguyên thủy giống nhau (ít nhất trong khoảng của chương trình). Bản thân các hệ số trong các ma trận nguyên thủy có thể nhưng cấu hình ma trận lại không thay đổi (hoặc không thay đổi thường xuyên như các hệ số). Cấu hình ma trận cho mỗi tầng có thể được xác định bởi các thông số đó như:

1. Số lượng các ma trận nguyên thủy trong tầng,
2. Thứ tự các kênh mà chúng thao tác,
3. Bậc độ lớn của các hệ số trong chúng,
4. Độ phân giải (bằng bit) cần thiết để thể hiện các hệ số, và
5. và các vị trí của các hệ số mà đồng nhất bằng 0.

Các thông số mà biểu thị cấu hình ma trận nguyên thủy như vậy có thể vẫn giữ nguyên trong khoảng của nhiều lần cập nhật ma trận hạt giống. Một hoặc nhiều trong số các thông số như vậy có thể cần được truyền qua dòng bit được mã hóa đến bộ giải mã để cho bộ giải mã hoạt động theo mong muốn. Vì các thông số cấu hình như vậy có thể không làm thay đổi thường xuyên bằng chính những lần cập nhật ma trận nguyên thủy, nên theo một số phương án cú pháp dòng bit được mã hóa độc lập biểu thị xem các thông số cấu hình ma trận có được truyền cùng với sự cập nhật đối với các hệ số ma trận của tập hợp các ma trận hạt giống không. Trái lại, ở TrueHD thông thường, các cập nhật ma trận mã hóa (được biểu thị bởi dòng bit được mã hóa) nhất thiết đi cùng với các cập nhật cấu hình. Theo các phương án được dự định của sáng chế, bộ giải mã duy trì và sử dụng các thông tin về cấu hình ma trận nhận được lần cuối nếu sự cập nhật chỉ được thu nhận cho các hệ số ma trận (ví dụ, không có sự cập nhật cấu hình ma trận).

Mặc dù dự kiến rằng việc lập ma trận nội suy sẽ thường cho phép tốc độ cập nhật ma trận hạt thấp, thì các phương án được dự định (trong đó cập nhật cấu hình ma trận có thể hoặc không thể đi cùng với mỗi lần cập nhật ma trận hạt) được mong đợi truyền hiệu quả các thông tin cấu hình và làm giảm hơn nữa tốc độ bit cần thiết để cập nhật các ma trận kết xuất. Theo các phương án được dự định, các thông số cấu hình có thể bao gồm các thông số gắn với mỗi ma trận hạt giống nguyên thủy, và/hoặc các thông số gắn với các ma trận delta được truyền.

Để giảm thiểu toàn bộ tốc độ bit được truyền, bộ mã hóa có thể tạo ra sự cân bằng giữa việc cập nhật cấu hình ma trận và sử dụng nhiều hơn một số bit vào các lần cập nhật hệ số ma trận trong khi duy trì cấu hình ma trận không thay đổi.

Việc ma trận nội suy có thể đạt được bằng cách truyền thông tin độ dốc để đi ngang qua từ một ma trận nguyên thủy cho kênh được mã hóa đến một ma trận khác mà hoạt động trên cùng một kênh. Độ dốc có thể được truyền dưới dạng tỉ lệ thay đổi các hệ số ma trận cho mỗi đơn vị truy cập ("AU"). Nếu m_1 và m_2 là các hệ số ma trận nguyên thủy cho các lần mà là K đơn vị truy cập riêng ra, sau đó độ dốc để nội suy từ m_1 đến m_2 có thể được định nghĩa là $\text{delta} = (m_2 - m_1)/K$.

Nếu các hệ số m_1 và m_2 bao gồm các bit có định dạng dưới đây: $m_1 = a.bcd\text{efg}$ và $m_2 = a.bcuvwx$, ở đó cả hai hệ số đều được biểu thị với số lượng cụ thể (mà có thể được biểu thị là "frac_bits") các bit có độ chính xác, sau đó độ dốc "delta" sẽ được biểu thị bởi một giá trị có định dạng 0.0000mnop (với độ chính xác cao hơn và các số 0 đứng đầu bổ sung được yêu cầu do đặc tả của các delta trên cơ sở mỗi AU). Độ chính xác bổ sung cần thiết để thể hiện độ dốc "delta" có thể được định nghĩa là "delta_precision". Nếu phương án của sáng chế bao gồm bước đưa mỗi giá trị delta một cách trực tiếp vào trong dòng bit được mã hóa, thì dòng bit được mã hóa sẽ cần phải bao gồm các giá trị có số lượng bit, "B," thỏa mãn phương trình: $B = \text{frac_bits} + \text{delta_precision}$. Rõ ràng không hiệu quả nếu truyền các số 0 đứng đầu sau chữ số thập phân. Vì vậy, theo một số phương án, những gì được mã hóa trong dòng bit được mã hóa (mà được phân phối cho bộ giải mã) là delta được chuẩn hóa (số nguyên) có dạng: mnopqr, mà được biểu diễn bằng delta_bits cộng với một bit tín hiệu. Các giá trị delta_bits và delta_precision có thể được truyền trong dòng bit được mã hóa dưới dạng một phần của thông tin cấu hình cho các ma trận delta. Theo các

phương án như vậy, bộ giải mã được tạo cấu hình để có được delta cần thiết trong trường hợp này dưới dạng

$$\text{delta} = (\text{delta được chuẩn hóa trong dòng bit}) * 2^{-(\text{frac_bits} + \text{delta_precision})}$$

Vì vậy, theo một số phương án, các giá trị nội suy có trong dòng bit đã mã hóa bao gồm các giá trị delta được chuẩn hóa có Y bit có độ chính xác (trong đó $Y = \text{frac_bits}$), và các giá trị chính xác. Các giá trị delta được chuẩn hóa biểu thị các phiên bản đã chuẩn hóa của các giá trị delta, trong đó các giá trị delta biểu thị các tỉ lệ thay đổi hệ số của các ma trận nguyên thủy, mỗi hệ số trong các ma trận nguyên thủy đều có Y bit có độ chính xác, và các giá trị chính xác biểu thị sự gia tăng về độ chính xác (ví dụ, “delta_precision”) cần thiết để biểu diễn các giá trị delta tương ứng với độ chính xác cần thiết để biểu diễn các hệ số của các ma trận nguyên thủy. Các giá trị delta có thể được tạo ra bằng cách chia tỷ lệ các giá trị delta đã chuẩn hóa bằng hệ số thang tỷ lệ mà tùy thuộc vào sự phân tích của các hệ số của các ma trận nguyên thủy và các giá trị chính xác.

Các phương án của sáng chế có thể được thực hiện ở phần cứng, phần sụn, hoặc phần mềm, hoặc tổ hợp của chúng (ví dụ, dưới dạng mảng logic có thể lập trình). Ví dụ, bộ mã hóa 40 hoặc 100, hoặc bộ giải mã 42 hoặc 102, hoặc các hệ thống phụ 47, 48, 60, và 61 của bộ giải mã 42, hoặc các hệ thống phụ 110-113 và 106-109 của bộ giải mã 102, có thể được thực hiện ở phần cứng hoặc phần sụn được lập trình phù hợp (hoặc theo cách khác được tạo cấu hình), ví dụ, dưới dạng bộ xử lý đa dụng được lập trình, bộ xử lý tín hiệu số, hoặc bộ vi xử lý. Trừ phi được định rõ khác, các thuật toán hoặc các quy trình được bao gồm như một phần của sáng chế vốn không có gắn với máy tính cụ thể hoặc thiết bị khác. Cụ thể, các máy đa dụng khác nhau có thể được sử dụng với các chương trình được viết theo các bậc lộ trong bản mô tả này, hoặc có thể thuận tiện hơn nếu xây dựng thiết bị chuyên dụng hơn (ví dụ, các mạch tích hợp) để thực hiện các bước cần thiết của phương pháp. Vì vậy, sáng chế có thể được thực hiện ở một hoặc nhiều chương trình máy tính thực thi trên một hoặc nhiều hệ thống máy tính có thể lập trình (ví dụ, hệ thống máy tính mà thực hiện bộ mã hóa 40 hoặc 100, hoặc bộ giải mã 42 hoặc 102, hoặc hệ thống phụ 47, 48, 60, và/hoặc 61 của bộ giải mã 42, hoặc các hệ thống phụ 110-113 và 106-109 của bộ giải mã 102), mỗi hệ thống bao gồm ít nhất một bộ xử lý, ít nhất một hệ thống lưu trữ dữ liệu (bao gồm bộ nhớ và/hoặc thành phần lưu trữ khả biến và không khả biến), ít nhất

một thiết bị hoặc cổng đầu vào, và ít nhất một thiết bị hoặc cổng đầu ra. Mã chương trình được áp dụng để nhập dữ liệu để thực hiện các chức năng được mô tả trong bản mô tả này và tạo ra các thông tin đầu ra. Thông tin đầu ra được áp dụng cho một hoặc nhiều thiết bị đầu ra, theo cách đã biết.

Mỗi chương trình như vậy có thể được thực hiện bằng ngôn ngữ máy tính mong muốn bất kỳ (bao gồm các ngôn ngữ máy móc, lắp ráp, hoặc quy trình cao cấp, logic, hoặc lập trình hướng đến đối tượng) để kết nối với hệ thống máy tính. Trong trường hợp bất kỳ, ngôn ngữ có thể là ngôn ngữ được biên soạn hoặc diễn giải.

Ví dụ, khi được thực hiện bởi các dãy lệnh phần mềm máy tính, thì các chức năng và bước khác nhau của các phương án của sáng chế có thể được thực hiện bởi các dãy lệnh phần mềm đa luồng chạy trong phần cứng xử lý tín hiệu số phù hợp, trong trường hợp đó các thiết bị, bước và chức năng khác nhau của các phương án có thể tương ứng với các phần của các lệnh phần mềm.

Tốt hơn là mỗi chương trình máy tính như vậy được lưu trữ trên hoặc cho xuống vào vật ghi hoặc thiết bị lưu trữ (ví dụ, bộ nhớ trạng thái rắn hoặc vật ghi, hoặc vật ghi từ tính hoặc quang học) có thể đọc được bởi máy tính có thể lập trình đa dụng hoặc chuyên dụng, để tạo cấu hình và hoạt động máy tính khi vật ghi hoặc thiết bị lưu trữ được đọc bởi hệ thống máy tính nhằm thực hiện các quy trình được mô tả ở bản mô tả này. Hệ thống theo sáng chế cũng có thể được thực hiện dưới dạng vật ghi lưu trữ đọc được bởi máy tính, tạo cấu hình với (ví dụ, lưu trữ) chương trình máy tính, ở đó vật ghi lưu trữ được tạo cấu hình như vậy khiến cho hệ thống máy tính hoạt động theo cách cụ thể và được xác định trước để thực hiện các chức năng được mô tả trong bản mô tả này.

Trong khi các phương án thực hiện đã được mô tả thông qua ví dụ, và theo các phương án cụ thể minh họa, thì cần hiểu rằng các phương án thực hiện của sáng chế không bị giới hạn ở các phương án được bộc lộ. Trái lại, sáng chế được dự định bao trùm các cải biến khác nhau và các bố trí tương tự mà sẽ hiển nhiên đối với người trung bình trong lĩnh vực kỹ thuật này. Do đó, phạm vi của các điểm yêu cầu bảo hộ đính kèm có được sự diễn giải rộng nhất sao cho bao gồm tất cả các cải biến và bố trí tương tự như vậy.

Yêu cầu bảo hộ

1. Phương pháp mã hóa chương trình âm thanh N kênh, trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, trong đó M nhỏ hơn hoặc bằng N, phương pháp này bao gồm các bước:

xác định tầng thứ nhất của các ma trận nguyên thủy $N \times N$ mà, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa, thực hiện phép trộn thứ nhất nội dung âm thanh của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó dạng trộn thứ nhất này gần đúng với $A(t_1)$, và trong đó ma trận nguyên thủy $N \times N$ được xác định là ma trận trong đó $N-1$ hàng chứa các phần tử ngoài đường chéo bằng 0 và các phần tử trên đường chéo có giá trị tuyệt đối bằng 1;

xác định các giá trị nội suy mà, với tầng thứ nhất gồm các ma trận nguyên thủy và hàm nội suy được xác định theo khoảng thời gian con, biểu thị dãy các tầng của các Ma trận nguyên thủy được cập nhật $N \times N$, sao cho mỗi tầng của các ma trận nguyên thủy được cập nhật, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa, thực hiện phép trộn cập nhật, gần với một thời điểm khác trong khoảng thời gian con, N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó mỗi dạng trộn được cập nhật đó gần đúng với ma trận trộn thay đổi theo thời gian, $A(t)$, ở thời gian trong khoảng thời gian con gần với dạng trộn được cập nhật; và

tạo ra dòng bit được mã hóa mà biểu thị nội dung âm thanh đã mã hóa, các giá trị nội suy, và tầng thứ nhất của các ma trận nguyên thủy.

2. Phương pháp theo điểm 1, trong đó mỗi ma trận nguyên thủy là ma trận nguyên thủy dạng đơn vị.

3. Phương pháp theo điểm 2, trong đó phương pháp này còn bao gồm bước tạo ra nội dung âm thanh được mã hóa bằng cách thực hiện các phép toán ma trận trên các mẫu của N kênh của chương trình, bao gồm áp dụng dãy các tầng ma trận cho các mẫu, trong đó mỗi tầng ma trận trong dãy là một tầng của các ma trận nguyên thủy, và dãy các tầng ma trận bao gồm tầng ma trận nghịch đảo thứ nhất là tầng của các dạng nghịch đảo của các ma trận nguyên thủy thuộc tầng thứ nhất.

4. Phương pháp theo điểm 2, trong đó phương pháp này còn bao gồm bước tạo ra nội dung âm thanh được mã hóa bằng cách thực hiện các phép toán ma trận trên mẫu của N kênh của chương trình, bao gồm áp dụng dãy các tầng ma trận cho các mẫu, trong đó mỗi tầng ma trận trong dãy là tầng của các ma trận nguyên thủy, và mỗi tầng ma trận trong dãy là nghịch đảo của tầng tương ứng trong số các tầng của ma trận nguyên thủy được cập nhật $N \times N$, và $N = M$, sao cho M kênh đầu ra là giống với N kênh của chương trình được phục hồi không mất dữ liệu.

5. Phương pháp theo điểm 1, trong đó tầng thứ nhất của các ma trận nguyên thủy thực hiện ma trận nguyên thủy hạt giống, và các giá trị nội suy biểu thị ma trận delta hạt giống cho ma trận nguyên thủy hạt giống.

6. Phương pháp theo điểm 4, trong đó ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, của nội dung âm thanh hoặc nội dung được mã hóa của chương trình với M_1 kênh loa đã được định rõ trong khoảng thời gian, trong đó M_1 là số nguyên nhỏ hơn M , và phương pháp này còn bao gồm bước:

xác định tầng thứ hai của ma trận nguyên thủy $M_1 \times M_1$ mà, khi được áp dụng cho các mẫu của M_1 kênh của nội dung âm thanh mã hóa ở mỗi thời điểm t trong khoảng thời gian thực hiện phép trộn giảm chương trình N kênh thành M_1 kênh loa, trong đó dạng trộn giảm này gần đúng với ma trận trộn thay đổi theo thời gian, $A_2(t)$.

7. Phương pháp phục hồi M kênh của chương trình âm thanh N kênh, trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, phương pháp này bao gồm các bước:

thu được dòng bit được mã hóa mà biểu thị nội dung âm thanh đã mã hóa, các giá trị nội suy, và tầng thứ nhất của ma trận nguyên thủy $N \times N$, trong đó ma trận nguyên thủy $N \times N$ được xác định là ma trận trong đó $N-1$ hàng chứa các phần tử ngoài đường chéo bằng 0 và các phần tử trên đường chéo với giá trị tuyệt đối bằng 1; và

thực hiện phép nội suy để xác định dãy các tầng của các ma trận nguyên thủy được cập nhật $N \times N$, từ các giá trị nội suy, tầng thứ nhất của các ma trận nguyên thủy, và hàm nội suy theo khoảng thời gian con, trong đó

tầng thứ nhất của ma trận nguyên thủy $N \times N$, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa của nội dung âm thanh đã mã hóa, thực hiện phép trộn thứ nhất của nội dung âm thanh của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó dạng trộn thứ nhất này gần đúng với $A(t_1)$, và

các giá trị nội suy, với tầng thứ nhất của ma trận nguyên thủy, và hàm nội suy, biểu thị dãy các tầng của ma trận nguyên thủy được cập nhật $N \times N$, sao cho mỗi tầng của ma trận nguyên thủy được cập nhật, khi được áp dụng cho các mẫu của N kênh tín hiệu được mã hóa của nội dung âm thanh đã mã hóa, thực hiện phép trộn được cập nhật, gắn với một thời điểm khác trong khoảng thời gian con, của N kênh tín hiệu được mã hóa thành M kênh đầu ra, trong đó mỗi dạng trộn được cập nhật đó gần đúng với ma trận trộn thay đổi theo thời gian, $A(t)$, ở thời gian trong khoảng thời gian con gắn với dạng trộn được cập nhật này.

8. Phương pháp theo điểm 7, trong đó mỗi ma trận nguyên thủy là ma trận nguyên thủy dạng đơn vị.

9. Phương pháp theo điểm 8, trong đó nội dung âm thanh đã mã hóa được tạo ra bằng cách thực hiện các phép toán ma trận trên mẫu của N kênh của chương trình, bao gồm áp dụng dãy các tầng ma trận cho các mẫu, trong đó mỗi tầng ma trận trong dãy là một tầng các ma trận nguyên thủy, và mỗi tầng ma trận trong dãy là nghịch đảo của tầng tương ứng của các tầng của ma trận nguyên thủy được cập nhật $N \times N$, và $N = M$, sao cho M kênh đầu ra là giống với N kênh của chương trình được phục hồi không mất dữ liệu.

10. Phương pháp theo điểm 9, trong đó ma trận trộn giảm thay đổi theo thời gian, $A_2(t)$, của nội dung âm thanh hoặc nội dung được mã hóa của chương trình với M_1 kênh loa cũng đã được định rõ trong khoảng thời gian, ở đó M_1 là số nguyên nhỏ hơn N , và phương pháp này còn bao gồm các bước:

nhận tầng thứ hai của ma trận nguyên thủy $M_1 \times M_1$; và

áp dụng tầng thứ hai của $M1 \times M1$ cho các mẫu của $M1$ kênh của nội dung âm thanh đã mã hóa ở mỗi thời điểm t trong khoảng thời gian để thực hiện phép trộn giảm chương trình N kênh thành $M1$ kênh loa, trong đó dạng trộn giảm này gần đúng với ma trận trộn thay đổi theo thời gian, $A2(t)$.

11. Phương pháp theo điểm 7, trong đó tầng thứ nhất của ma trận nguyên thủy thực hiện ma trận nguyên thủy hạt giống, và các giá trị nội suy biểu thị ma trận delta hạt giống cho ma trận nguyên thủy hạt giống này.

12. Phương pháp theo điểm 7, trong đó phương pháp này còn bao gồm bước:

áp dụng ít nhất một trong các tầng của ma trận nguyên thủy được cập nhật $N \times N$ cho các mẫu nội dung âm thanh được mã hóa, bao gồm bằng cách áp dụng ma trận nguyên thủy hạt giống và ma trận delta hạt giống một cách riêng rẽ cho các mẫu nội dung âm thanh được mã hóa để tạo ra các mẫu được biến đổi, và kết hợp tuyến tính các mẫu được biến đổi theo hàm nội suy, nhờ đó tạo ra các mẫu đã phục hồi biểu thị các mẫu của M kênh của chương trình âm thanh N kênh.

13. Phương pháp theo điểm 7, trong đó hàm nội suy là không đổi trong một số khoảng của dòng bit được mã hóa, và mỗi tầng được cập nhật mới nhất trong các tầng của ma trận nguyên thủy $N \times N$ đã cập nhật được cập nhật bằng phép nội suy chỉ trong khoảng của dòng bit được mã hóa trong đó hàm nội suy thực chất không phải là bất biến.

14. Phương pháp theo điểm 7, trong đó các giá trị nội suy bao gồm các giá trị delta đã chuẩn hóa có thể biểu diễn với Y bit, chỉ báo về số lượng bit có độ chính xác, và các giá trị chính xác, trong đó các giá trị delta được chuẩn hóa biểu thị các phiên bản đã chuẩn hóa của các giá trị delta, các giá trị delta biểu thị các tỉ lệ thay đổi hệ số của các ma trận nguyên thủy, và các giá trị chính xác biểu thị sự gia tăng về độ chính xác cần thiết để biểu diễn các giá trị delta tương ứng với độ chính xác cần thiết để biểu diễn các hệ số của các ma trận nguyên thủy.

15. Phương pháp theo điểm 14, trong đó các giá trị delta được tạo ra bằng cách chia tỷ lệ các giá trị delta được chuẩn hóa bằng hệ số thang tỷ lệ mà tùy thuộc vào sự

phân tích các hệ số của các ma trận nguyên thủy và các giá trị chính xác.

16. Phương pháp theo điểm 7, trong đó phương pháp này còn bao gồm bước:

trích từ kiểm tra từ dòng bit được mã hóa, và

xác minh xem các kênh của đoạn chương trình âm thanh có được phục hồi đúng không, bằng cách so sánh từ kiểm tra thứ hai được tạo ra từ các mẫu âm thanh được tạo ra bởi hệ thống phụ nhân ma trận với từ kiểm tra được trích từ dòng bit được mã hóa.

17. Bộ mã hóa âm thanh được tạo cấu hình để mã hóa chương trình âm thanh N kênh, trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, trong đó M nhỏ hơn hoặc bằng N, và trong đó bộ mã hóa này được tạo cấu hình để thực hiện phương pháp theo điểm 1.

18. Bộ giải mã được tạo cấu hình để thực hiện việc phục hồi chương trình âm thanh N kênh, trong đó chương trình được định rõ trong một khoảng thời gian, khoảng thời gian này bao gồm khoảng thời gian con từ thời điểm t_1 đến thời điểm t_2 , và ma trận trộn thay đổi theo thời gian, $A(t)$, của N kênh tín hiệu được mã hóa thành M kênh đầu ra đã được định rõ trong một khoảng thời gian, và trong đó bộ giải mã này được tạo cấu hình để thực hiện phương pháp theo điểm 7.

19. Vật ghi bất biến có thể đọc được bằng máy tính bao gồm chương trình của các lệnh, mà, khi được thực thi bởi bộ mã hóa âm thanh, khiến cho bộ mã hóa âm thanh thực hiện phương pháp theo điểm 1.

20. Vật ghi bất biến có thể đọc được bằng máy tính bao gồm chương trình của các lệnh, mà, khi được thực thi bởi bộ giải mã, khiến cho bộ giải mã thực hiện phương pháp theo điểm 7.

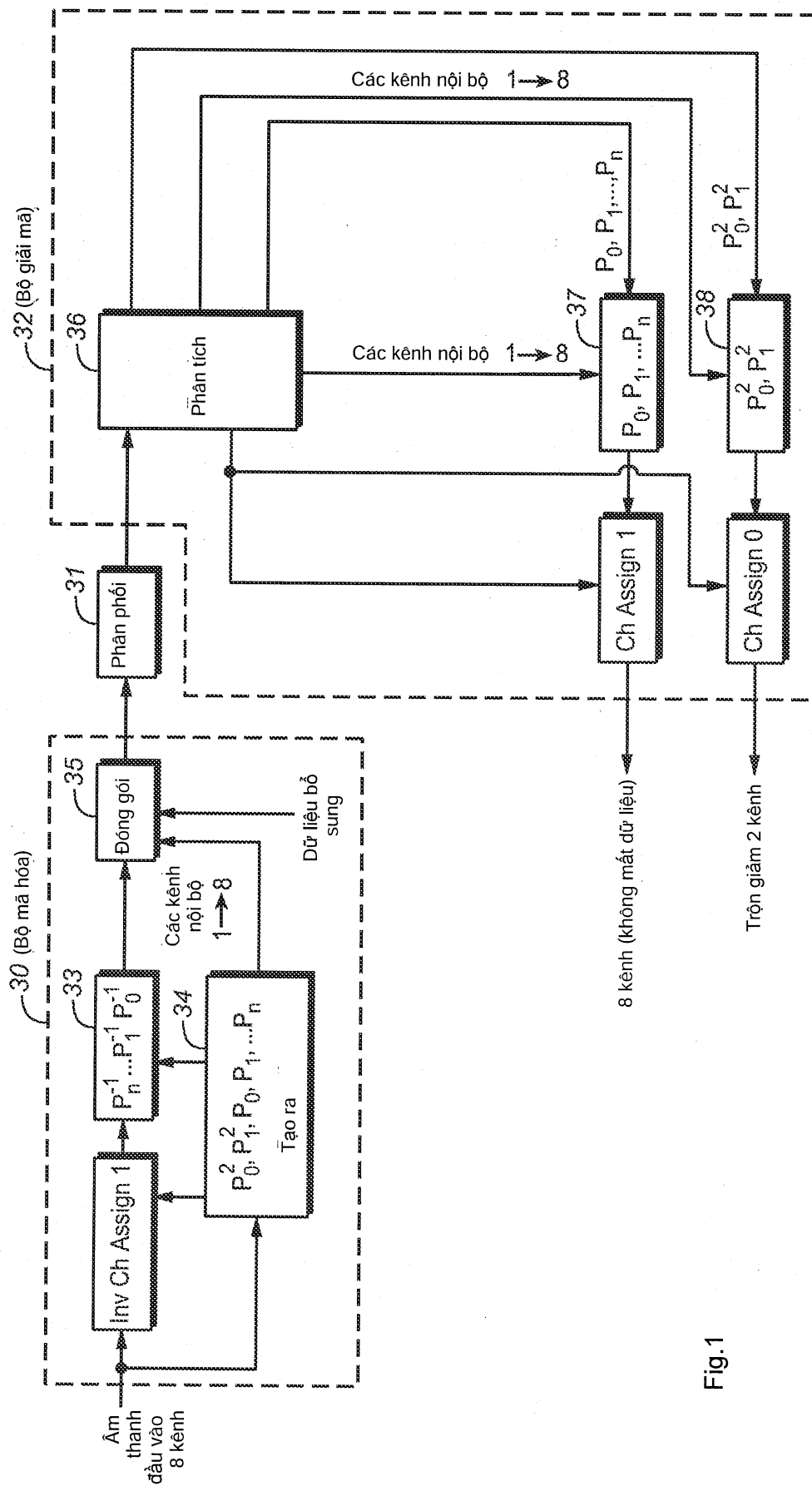


Fig.1

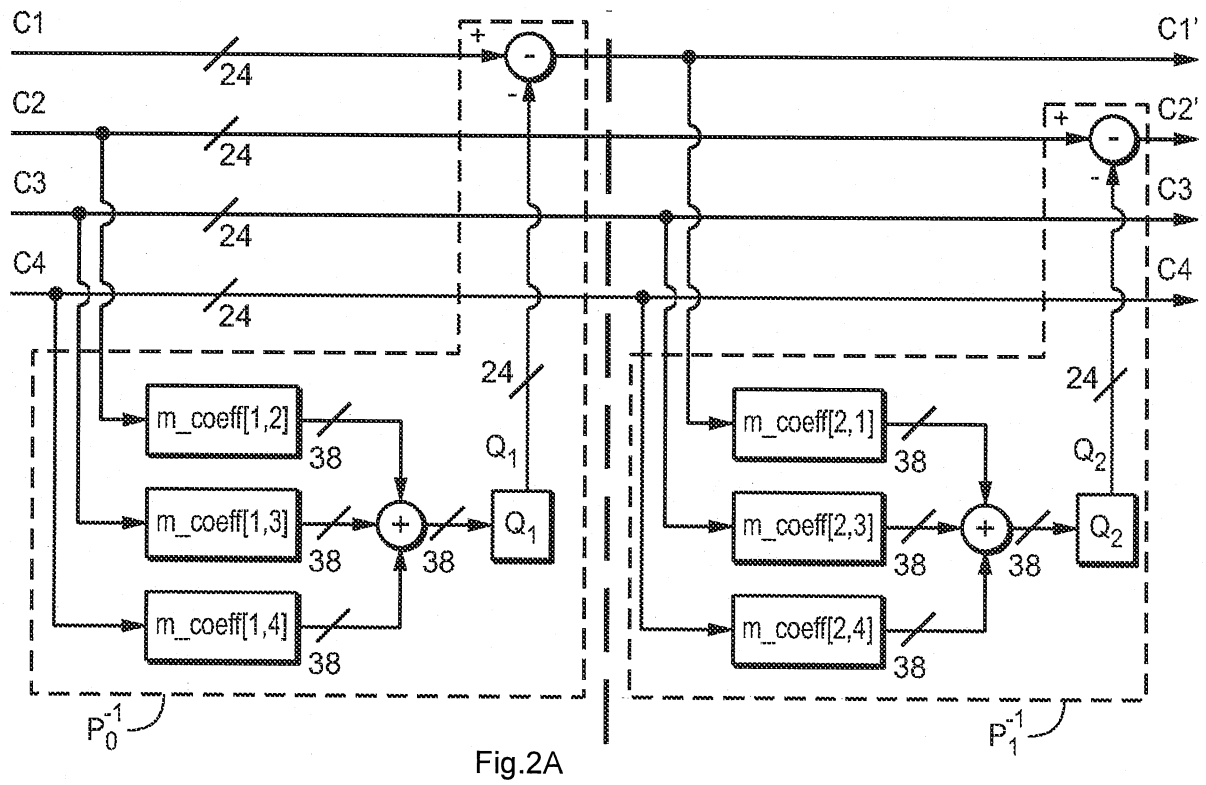


Fig.2A

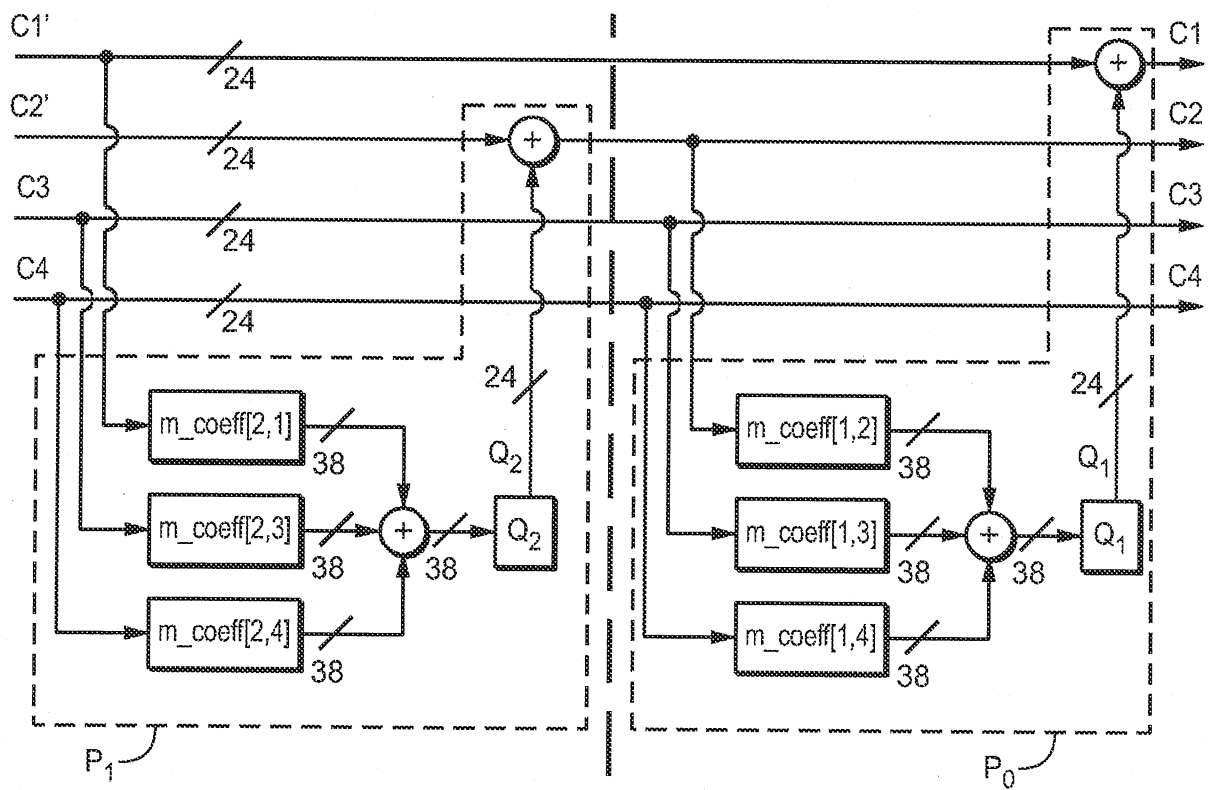


Fig.2B

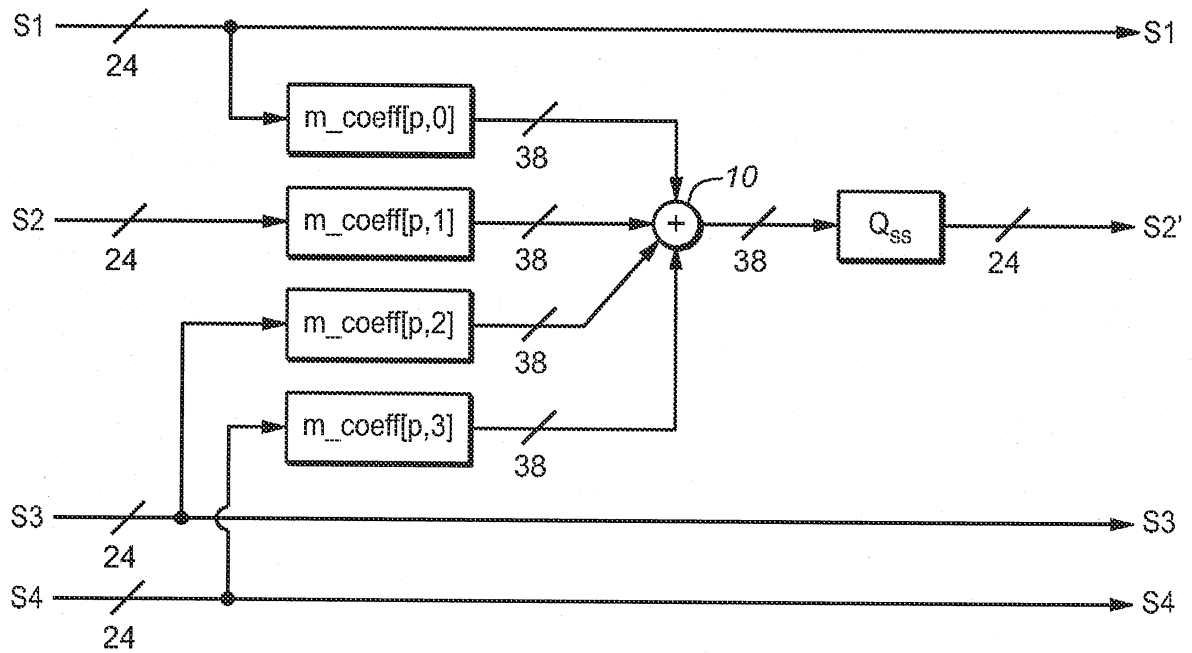


FIG. 3

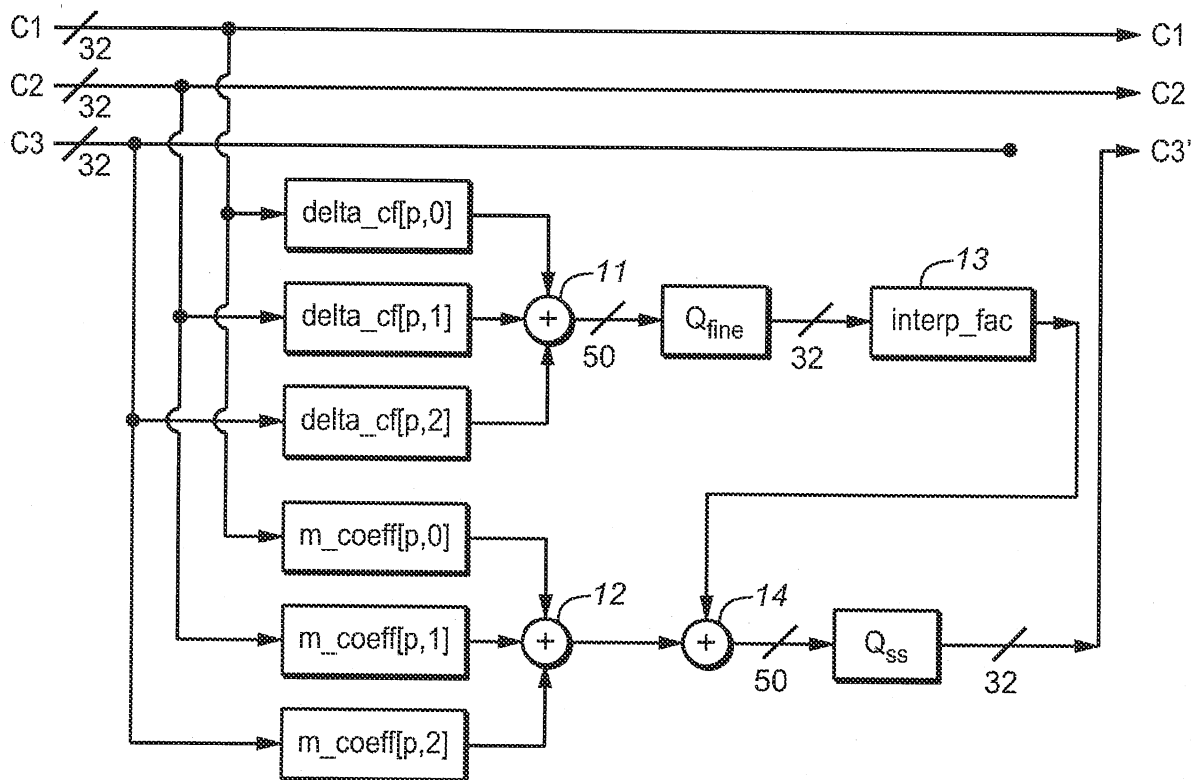


FIG. 4

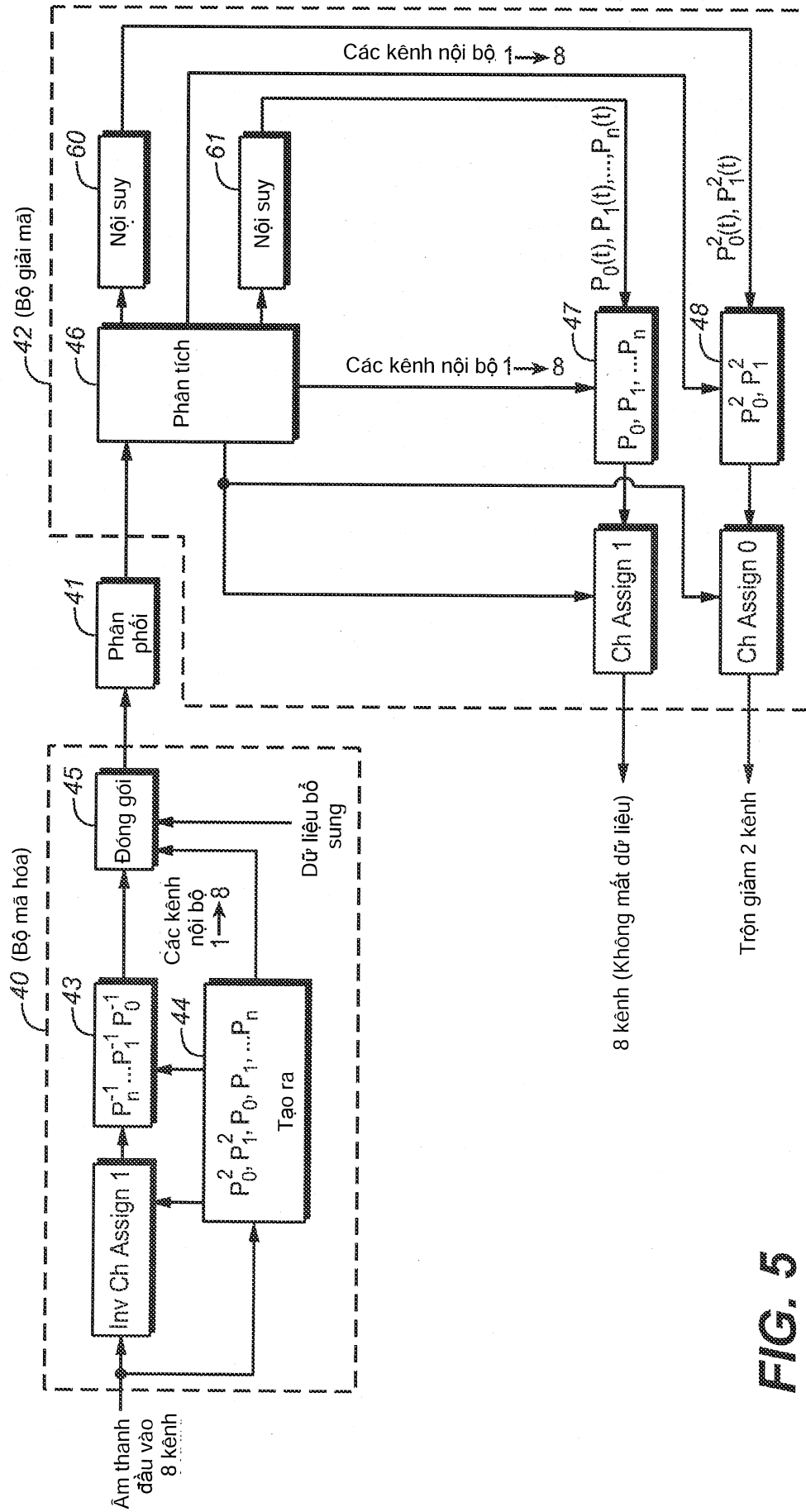
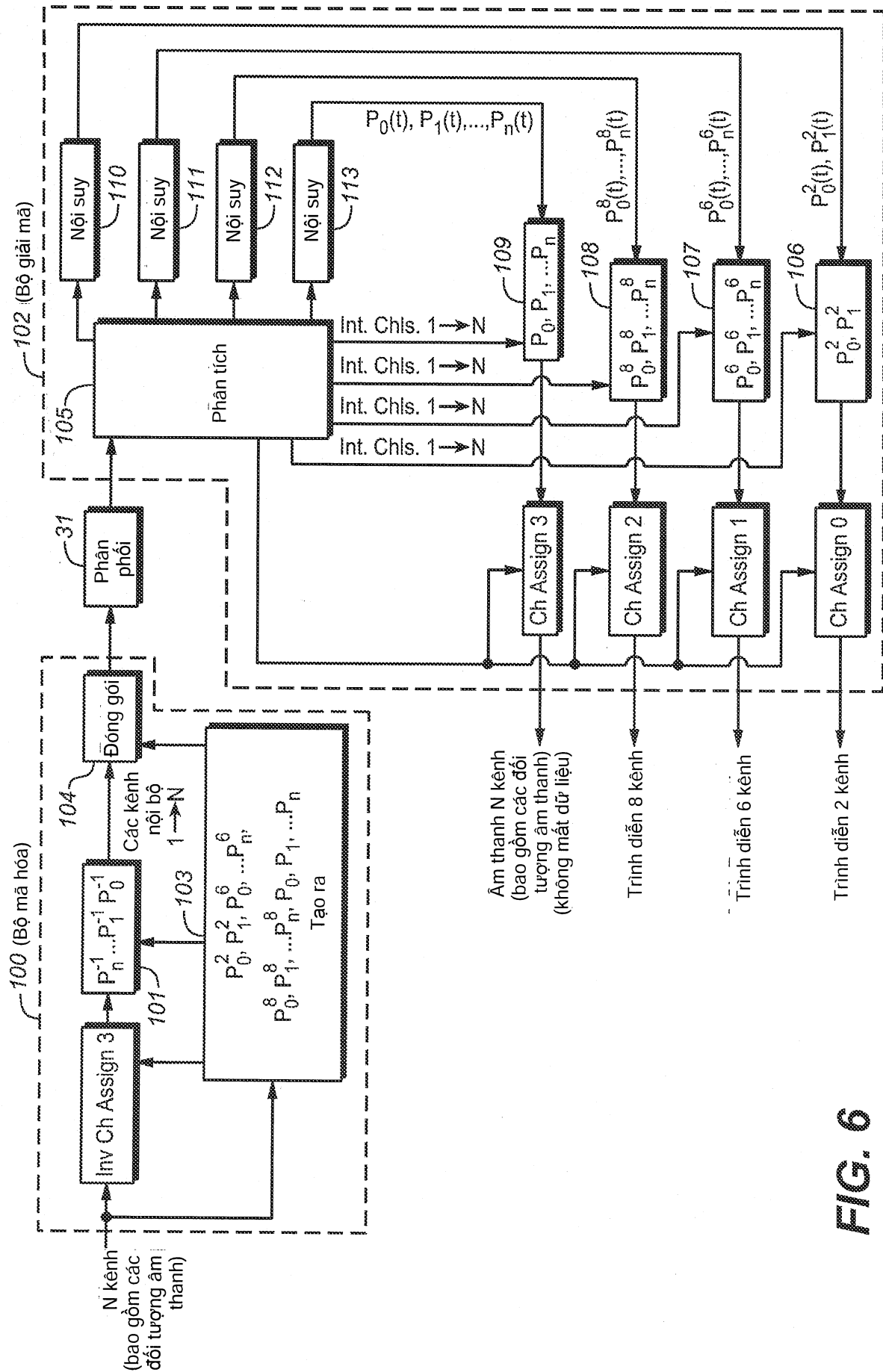


FIG. 5



694

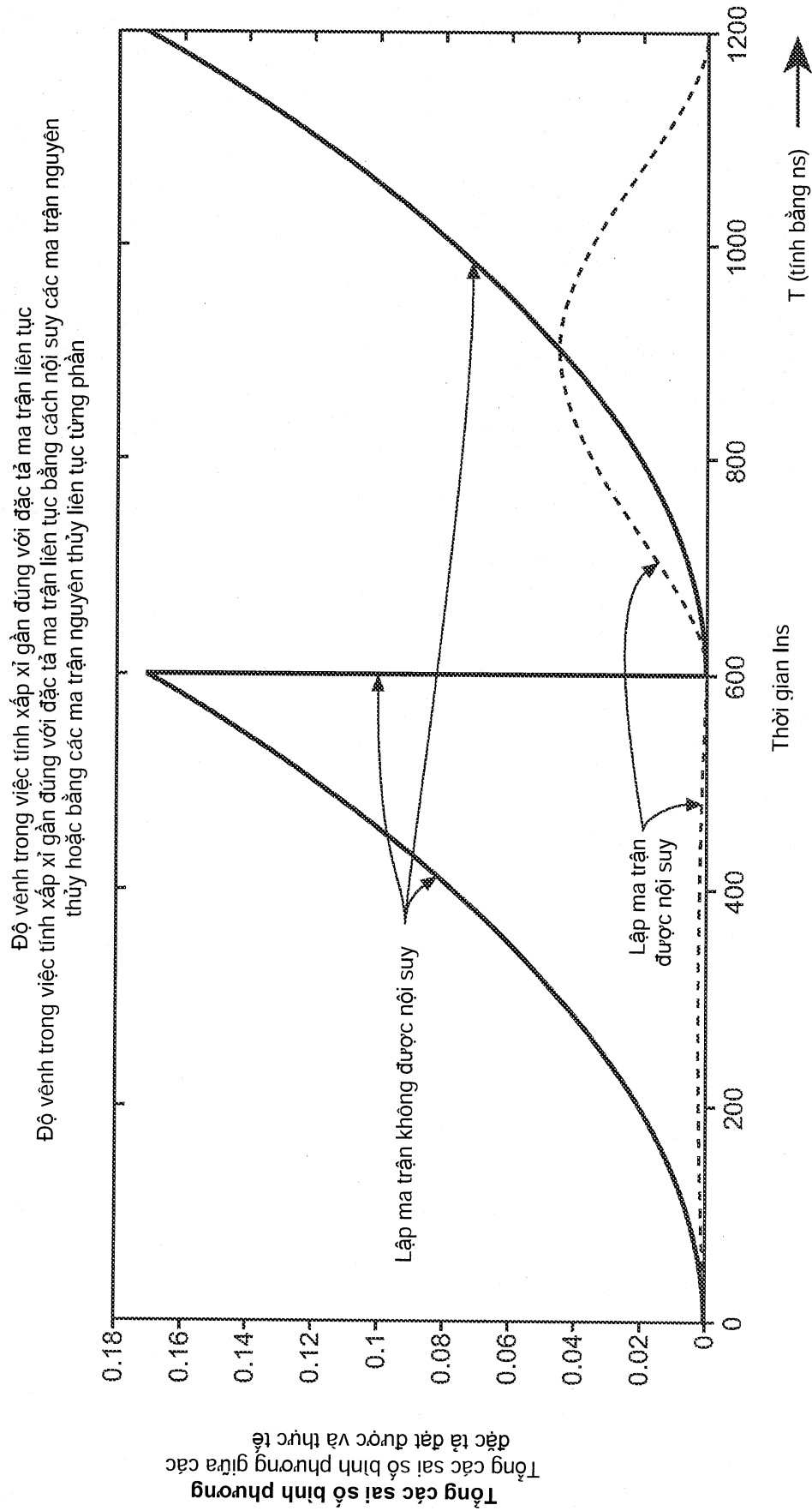


FIG. 7