



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

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11)
CỤC SỞ HỮU TRÍ TUỆ



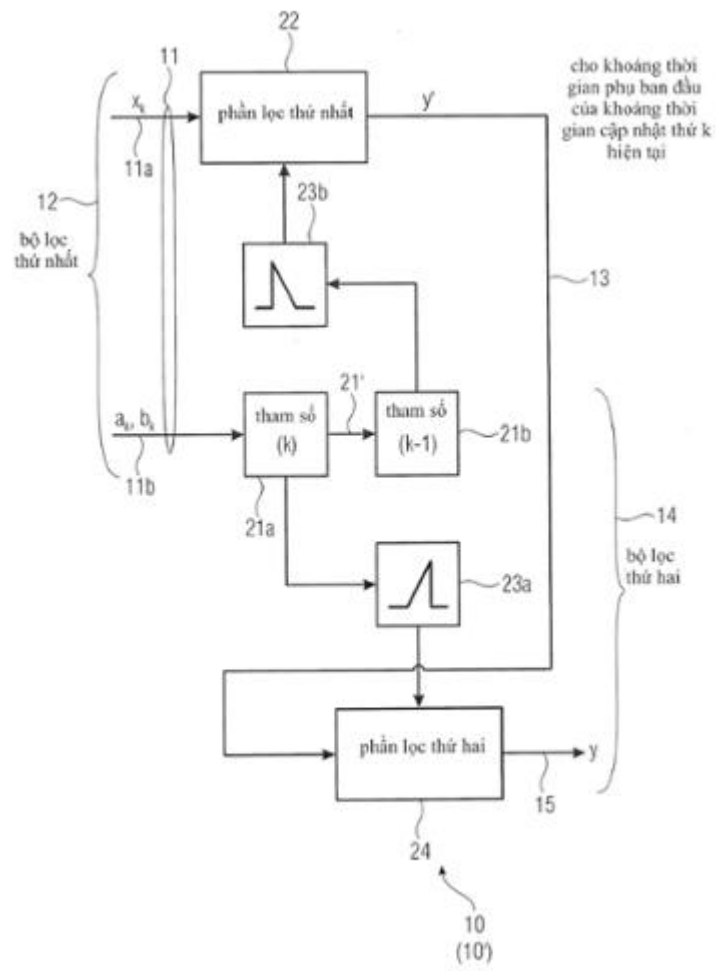
1-0042105

(51)^{2020.01} **G10L 19/26; G10L 19/04; H04N 19/117; (13) B**
H03H 17/02; G10L 19/00; G10L 19/09

-
- (21) 1-2020-03253 (22) 09/11/2018
(86) PCT/EP2018/080837 09/11/2018 (87) WO 2019/092220 16/05/2019
(30) 17201105.8 10/11/2017 EP
(45) 25/12/2024 441 (43) 25/09/2020 390
(73) Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung e. V. (DE)
Hansastrasse 27c, 80686 Muenchen, Germany
(72) MARKOVIC, Goran (DE); RAVELLI, Emmanuel (FR); DIETZ, Martin (DE);
GRILL, Bernhard (DE).
(74) CÔNG TY LUẬT TRÁCH NHIỆM HỮU HẠN AMBYS HÀ NỘI (AMBYS
HANOI)
-

(54) HỆ THỐNG VÀ PHƯƠNG PHÁP LỌC TÍN HIỆU THÔNG TIN ĐẦU VÀO

(57) Sáng chế đề cập đến phương pháp và hệ thống lọc tín hiệu thông tin đầu vào. Hệ thống có thể bao gồm: bộ lọc thứ nhất (12) lọc tín hiệu đầu vào tại khoảng thời gian phụ ban đầu (τ_l) trong khoảng thời gian cập nhật hiện tại (T) theo các tham số liên quan đến khoảng thời gian cập nhật trước đó, các tham số được định tỉ lệ theo hệ số định tỉ lệ thứ nhất (s_{k-1}) thay đổi về 0; bộ lọc thứ hai (14) lọc tín hiệu đầu vào bộ lọc thứ hai (13), dựa trên đầu ra của bộ lọc thứ nhất, tại khoảng thời gian phụ ban đầu (τ_l), theo các tham số liên quan đến khoảng thời gian cập nhật hiện tại (T), các tham số được định tỉ lệ theo hệ số định tỉ lệ thứ hai (s_k) thay đổi từ 0 hoặc giá trị gần 0, đến giá trị xa 0 hơn.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến hệ thống và phương pháp lọc tín hiệu (ví dụ, lọc sau và/hoặc lọc trước LTP).

Tình trạng kỹ thuật của sáng chế

Âm thanh và giọng nói nói chung là các tín hiệu thay đổi theo thời gian. Vì những thay đổi tương đối chậm nên chúng thường được coi là gần như đứng yên trong khoảng thời gian ngắn. Các tham số bộ lọc thích ứng (ví dụ: mã hóa dự báo tuyến tính (linear predictive coding-LPC) hoặc bộ lọc sau dài hạn (long-term post filter-LTP) được sử dụng để xử lý tín hiệu âm thanh/giọng nói được cập nhật một lần trên mỗi khung và giữ nguyên trong thời lượng khung, khung thường có độ dài từ 2 đến 40 mili giây. Việc lọc như vậy có hiệu quả thay đổi theo thời gian và do đó nói chung tạo ra sự không ổn định và không liên tục ngay cả khi lọc với các tham số bộ lọc bị đóng băng cũng không [1].

Cách tiếp cận nhiều xuyên kênh đã biết. Cách tiếp cận nhiều xuyên kênh có thể được tóm tắt như sau:

- lọc một phần của tín hiệu với tập hợp tham số c_0 tạo phần được lọc thứ nhất
- lọc phần tương tự của tín hiệu với tập hợp tham số c_1 tạo ra phần được lọc thứ hai; và
- thực hiện làm nhiều xuyên kênh phần đã lọc thứ nhất và thứ hai.

Phương thức làm nhiều xuyên kênh được sử dụng trong số các phương pháp khác trong tài liệu [9], [10], [11] và [12].

- Một cách tiếp cận khác để loại bỏ sự không liên tục là sử dụng bộ lọc LP như trong tài liệu [13].

Bộ lọc đáp ứng xung vô hạn (infinite impulse response-IIR) có thể được trình bày ở dạng bậc thang mạng trong tài liệu [7]. Phép nội suy các bộ lọc IIR ổn định ở dạng bậc thang lưới sẽ tạo ra bộ lọc IIR thay đổi theo thời gian ổn định theo tài liệu [8]. Do đó, có thể khái quát hóa phép nội suy của các hệ số phản xạ từ tài liệu [6] trong trường

hợp chung của các bộ lọc IIR. Tuy nhiên, cách tiếp cận như vậy sẽ quá phức tạp đối với lọc LTP vì số lượng hệ số phản xạ khác không sẽ bằng độ trễ tần số cơ bản (ví dụ: nếu được sử dụng để làm nhẵn các điểm gián đoạn LTP trong tài liệu [3], điều này sẽ dẫn đến các yêu cầu lọc lớn hơn 250).

Độ phức tạp là khía cạnh rất quan trọng trong bộ mã hóa-giải mã thời gian thực và nên sử dụng phương pháp để tránh sự gián đoạn trong quá trình lọc thay đổi theo thời gian với độ phức tạp ít nhất.

Kỹ thuật có độ phức tạp thấp được ưu tiên để thực hiện các hoạt động lọc.

Tài liệu tham khảo

- [1] A. T. Hill and A. Ilchmann, "Exponential stability of time-varying linear systems," IMA J Numer Anal, pp. 865-885, 2011.
- [2] 3GPP TS 26.090, Adaptive Multi-Rate (AMR) speech codec; Transcoding functions.
- [3] 3GPP TS 26.445, Codec for Enhanced Voice Services (EVS); Detailed algorithmic description.
- [4] 3GPP TS 26.190, Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Transcoding functions.
- [5] 3GPP TS 26.290, Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec; Transcoding functions.
- [6] B. Edler, C. Faller and G. Schuller, "Perceptual Audio Coding Using a Time-Varying Linear Pre- and Post-Filter," in AES 109th Convention, Los Angeles, 2000.
- [7] A. Gray and J. Markel, "Digital lattice and ladder filter synthesis," IEEE Transactions on Audio and Electroacoustics, vol. vol. 21, no. no. 6, pp. 491-500, 1973.
- [8] M. P. Lamoureux, S. Ismail and G. F. Margrave, "Stability of time variant filters," CREWES Research Report - Volume 19, 2007.
- [9] P. J. Wilson and H. Chhatwal, "Adaptive Transform Coder Having Long Term Predictor". US Patent 5,012,517, 30 Apr 1991.

- [10] M. Tsushima, Y. Nakatoh and T. Norimatus, "Apparatus for expanding speech bandwidth". EP Patent 0 732 687 B2, 18 Sep 1996.
- [11] A. John Robinson, "Low Bit Rate Audio Coder And Decoder Operating in A Transform Domain Using Vector Quantization". US Patent 5,999,899, 7 Dec 1999.
- [12] J. Thyssen, C. C Lee and J.-H. Chen, "Method And Apparatus To Eliminate Discontinuities In Adaptively Filtered Signals". US Patent 7,353,168 B2, 28 Jun 2002.
- [13] E. Ravelli, M. Jander, G. Pietrzyk, M. Dietz and M. Gayer, "Method and apparatus for processing an audio signal, audio decoder, and audio encoder". EP Patent 2980796 A1, 28 Jul 2014.
- [14] E. Ravelli, C. Helmrich, G. Markovic, M. Neusinger, M. Jander, M. Dietz and S. Disch, "Apparatus and method for processing an audio signal using a harmonic post-filter". EP Patent 2980799 A1, 28 Jul 2014.
- [15] ITU-T Recommendation G.718, Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s, 2008.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế đề xuất kỹ thuật có độ phức tạp thấp để tránh sự không liên tục, ví dụ: khi các tham số lọc đáp ứng xung vô hạn (Infinite Impulse Response-IIR) thay đổi trong các khung liên tiếp.

Theo các ví dụ, sáng chế đề xuất hệ thống lọc tín hiệu thông tin đầu vào, được chia thành các khoảng cập nhật khác nhau, theo các tham số thay đổi theo các khoảng cập nhật, để thu được tín hiệu đầu ra được lọc, hệ thống bao gồm:

bộ lọc thứ nhất để lọc tín hiệu đầu vào bộ lọc thứ nhất ít nhất tại khoảng thời gian phụ ban đầu trong khoảng thời gian cập nhật hiện tại để thu được tín hiệu đầu ra bộ lọc thứ nhất, theo các tham số liên quan đến khoảng thời gian cập nhật trước đó, bộ lọc thứ nhất được tạo cấu hình để thay đổi các tham số cùng với ít nhất là khoảng thời gian phụ ban đầu từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn; và

bộ lọc thứ hai để lọc tín hiệu đầu vào bộ lọc thứ hai, tại khoảng thời gian ban đầu, theo các tham số liên quan đến khoảng thời gian cập nhật hiện tại để thu được tín hiệu đầu ra bộ lọc thứ hai, bộ lọc thứ hai được tạo cấu hình để thay đổi các tham số theo ít nhất là khoảng thời gian phụ ban đầu từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn,

trong đó tín hiệu đầu vào bộ lọc thứ nhất dựa trên tín hiệu thông tin đầu vào, tín hiệu đầu ra bộ lọc thứ nhất là tín hiệu trung gian, tín hiệu đầu vào bộ lọc thứ hai dựa trên tín hiệu trung gian và tín hiệu đầu ra được lọc dựa trên tín hiệu đầu ra bộ lọc thứ hai.

Theo đó, sự không liên tục được giảm và/hoặc tránh và độ phức tạp giảm, ví dụ, so với kỹ thuật làm nhiễu xuyên kênh. Không cần thực hiện hai hoạt động lọc khác nhau và sau đó, để làm nhiễu xuyên kênh hai tín hiệu được lọc: đơn giản, hai hoạt động lọc được thực hiện, do đó làm giảm sự cần thiết phải thực hiện tính toán.

Theo các ví dụ, sáng chế đề xuất bộ lọc thứ ba để lọc tín hiệu thông tin đầu vào, tại một khoảng thời gian phụ tiếp theo trong khoảng thời gian cập nhật hiện tại sau khoảng thời gian phụ ban đầu theo các tham số liên quan đến khoảng thời gian cập nhật hiện tại.

Theo đó, ngoài khoảng thời gian phụ ban đầu, khoảng thời gian cập nhật hiện tại có thể hoạt động bằng cách sử dụng các tham số phù hợp nhất với nó.

Theo các ví dụ, sáng chế đề xuất bộ chọn được tạo cấu hình để kiểm tra xem khoảng cách giữa các tham số cho khoảng thời gian cập nhật hiện tại là 0 hay trong ngưỡng thứ nhất, để lọc tín hiệu thông tin đầu vào ít nhất là trong khoảng thời gian phụ ban đầu sử dụng bộ lọc thứ ba.

Theo các ví dụ, bộ lọc thứ ba được tạo cấu hình để duy trì trạng thái lọc tương tự dọc theo khoảng thời gian tiếp theo và hoặc để duy trì tham số bộ lọc không đổi.

Theo các ví dụ, ít nhất một trong số bộ lọc thứ nhất, thứ hai và thứ ba hoạt động như bộ lọc LTP dài hạn.

Theo các ví dụ, ít nhất một trong số bộ lọc thứ nhất, thứ hai và thứ ba có hàm chuyển bao gồm tử số và mẫu số, trong đó tử số bao gồm giá trị độ khuếch đại được biểu thị bởi thông tin độ khuếch đại, và trong đó mẫu số bao gồm phần nguyên của độ

trễ tần số cơ bản được biểu thị bởi thông tin độ trễ tần số cơ bản và bộ lọc đa chạm phụ thuộc vào phần thập phân của độ trễ tần số cơ bản.

Theo các ví dụ, các tham số của ít nhất một trong các bộ thứ nhất, thứ hai và thứ ba thu được từ thông tin điều hòa, thông tin độ khuếch đại, thông tin độ trễ tần số cơ bản, phần nguyên của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào và/hoặc phần thập phân của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào.

Theo các ví dụ, các tham số của ít nhất một trong số bộ lọc thứ nhất, thứ hai và/hoặc thứ ba là các tham số của bộ lọc được chọn trong số ít nhất một hoặc kết hợp của bộ lọc mã hóa dự báo tuyến tính (linear predictive coding-LPC), bộ lọc đáp ứng xung vô hạn (infinite impulse response-IIR) và/hoặc bộ lọc đáp ứng xung hữu hạn (finite impulse response-FIR).

Theo các ví dụ, bộ lọc thứ nhất được tạo cấu hình để định tỉ lệ các tham số liên quan đến khoảng thời gian cập nhật trước bởi hệ số tỉ lệ thứ nhất thay đổi về 0 cùng với ít nhất là khoảng thời gian phụ ban đầu và/hoặc bộ lọc thứ hai được tạo cấu hình để định tỉ lệ các tham số liên quan đến khoảng thời gian cập nhật hiện tại theo hệ số tỉ lệ thứ hai thay đổi từ 0 thành giá trị khác 0 (hoặc từ giá trị gần 0 đến giá trị thứ hai cách xa 0 hơn so với giá trị gần 0) cùng với ít nhất là khoảng thời gian phụ ban đầu.

Theo các ví dụ, hệ số tỉ lệ thứ nhất và hệ số tỉ lệ thứ hai là các giá trị không âm bổ sung cho nhau với giá trị lớn hơn 0.

Theo các ví dụ, hệ số định tỉ lệ thứ nhất là thay đổi về 0 về phía cực trị cuối cùng của ít nhất là khoảng thời gian phụ ban đầu và/hoặc hệ số định tỉ lệ thứ hai là thay đổi từ 0 từ cực trị ban đầu của khoảng thời gian cập nhật hiện tại đến giá trị khác 0 (hoặc từ giá trị gần 0 đến giá trị thứ hai cách xa 0 hơn giá trị gần 0).

Theo các ví dụ, sáng chế đề xuất bộ lọc thứ tư được tạo cấu hình để lọc tín hiệu thông tin đầu vào, ít nhất là tại khoảng thời gian phụ ban đầu, sử dụng các tham số thu được bằng cách nội suy các tham số liên quan đến khoảng thời gian cập nhật hiện tại và các tham số liên quan đến khoảng thời gian cập nhật trước đó.

Theo các ví dụ, sáng chế đề xuất bộ chọn được tạo cấu hình để kiểm tra xem khoảng cách giữa các tham số cho khoảng thời gian cập nhật hiện tại là nằm trong ngưỡng thứ

hai hay không để lọc tín hiệu thông tin đầu vào ít nhất là trong khoảng thời gian phụ ban đầu sử dụng bộ lọc thứ tư.

Hệ thống có thể còn được tạo cấu hình để chủ động đặt ngưỡng thứ hai trên cơ sở các giá trị liên quan đến tín hiệu.

Theo các ví dụ, hệ thống có thể được tạo cấu hình để:

đặt ngưỡng thứ hai là ngưỡng khoảng cách độ trễ tần số cơ bản bằng với mức tối thiểu giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó,

để sử dụng bộ lọc thứ tư khi khoảng cách giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó nhỏ hơn ngưỡng khoảng cách độ trễ tần số cơ bản; và/hoặc

để sử dụng các bộ lọc thứ nhất và thứ hai khi khoảng cách giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó lớn hơn ngưỡng khoảng cách độ trễ tần số cơ bản.

Theo các ví dụ, hệ thống có thể được tạo cấu hình để:

sử dụng điều kiện liên quan đến độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó,

để sử dụng bộ lọc thứ tư khi cả độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó khác với 0 và/hoặc

để sử dụng bộ lọc thứ nhất và thứ hai khi ít nhất một độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó là không.

Theo các ví dụ, bộ lọc thứ nhất là để cung cấp tín hiệu đầu ra bộ lọc thứ nhất ở dạng

$$y'[n] = x[n] + s_{k-1}[n] \left(\sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y'[n-j] \right), kT \leq n < kT + T_l$$

trong đó

$s_{k-1}[n]$ thay đổi về giá trị gần 0 khi n tăng

và bộ lọc thứ hai là để cung cấp tín hiệu đầu ra bộ lọc thứ hai ở dạng

$$y[n] = y'[n] + s_k[n] \left(\sum_{i=0}^P b_{k,i} y'[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j] \right), kT \leq n < kT + T_l$$

trong đó

$s_k[n]$ thay đổi từ giá trị a gần về 0

giá trị a khác không khi n tăng

Trong đó T là khoảng thời gian cập nhật thứ k hiện tại, T_l là khoảng thời gian phụ ban đầu, n là khoảng khác, $x[n]$ là tín hiệu thông tin đầu vào, $b_{k-1,i}$ và $a_{k-1,j}$ là các tham số liên quan đến khoảng thời gian cập nhật thứ $(k-1)$ trước đó, $a_{k,j}$ và $b_{k,i}$ là các tham số liên quan đến khoảng thời gian cập nhật thứ k hiện tại, và P và Q liên quan đến loại bộ lọc.

Theo các ví dụ, bộ lọc thứ nhất được tạo cấu hình để cung cấp tín hiệu đầu ra bộ lọc thứ nhất ở dạng

$$\widehat{x_{ltpf}}'(n) = \hat{x}(n) - \left(1 - \frac{n}{\frac{N_F}{4}}\right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x_{ltpf}}' \left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k\right) \right]$$

$for\ n = 0.. \frac{N_F}{4}$

và bộ lọc thứ hai được tạo cấu hình để cung cấp tín hiệu đầu ra đã lọc 13 ở dạng

$$\widehat{x_{ltpf}}(n) = \widehat{x_{ltpf}}'(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x_{ltpf}}'(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}} \left(n - p_{int} + \frac{L_{den}}{2} - k\right) \right]$$

$for\ n = 0.. \frac{N_F}{4}$

trong đó $\frac{N_F}{4}$ là chiều dài của khoảng thời gian phụ ban đầu, $\hat{x}(n)$ là tín hiệu thông tin đầu vào, $\widehat{x_{ltpf}}'$ là tín hiệu trung gian, $\widehat{x_{ltpf}}(n)$ là tín hiệu đầu ra đã lọc, n là khoảng khác, p_{int}^{mem} và p_{fr}^{mem} là tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khoảng thời gian cập nhật trước đó, p_{int} và p_{fr} là tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khoảng

thời gian cập nhật hiện tại, $c_{num}(k)$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật hiện tại, $c_{den}(k, p_{fr})$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật được xác định và dựa trên phần thập phân của tần số cơ bản, $c_{num}^{mem}(k)$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật trước đó, $c_{den}^{mem}(k, p_{fr}^{mem})$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật trước đó và dựa trên phần thập phân của tần số cơ bản, L_{den} và L_{num} được cố định và/hoặc dựa trên tốc độ lấy mẫu của tín hiệu đầu vào.

Theo các ví dụ, độ dài thời gian của khoảng thời gian phụ ban đầu nằm trong khoảng từ 5% đến 40% độ dài thời gian của khoảng thời gian cập nhật hiện tại.

Theo các ví dụ, hệ thống được tạo cấu hình để kiểm tra độ khuếch đại g_k của khung thứ k hiện tại và độ khuếch đại g_{k-1} của khung thứ $(k-1)$ trước đó, sao cho:

- Nếu $g_{k-1}=0$ và $g_k=0$, thì không lọc thứ nhất, thứ hai cũng như thứ ba ;
và/hoặc
- Nếu $g_{k-1}=0$ và $g_k \neq 0$, thì
 - bước lọc thứ nhất được khởi kích hoạt
 - có bước lọc thứ hai trong ít nhất khoảng thời gian phụ ban đầu;
 - có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo;
 và/hoặc
- Nếu $g_{k-1} \neq 0$ và $g_k \neq 0$, thì
 - có bước lọc thứ nhất trong ít nhất khoảng thời gian phụ ban đầu;
 - bước lọc thứ hai được khởi kích hoạt
 - lọc thứ ba được khởi kích hoạt; và/hoặc
- Nếu $g_{k-1} \neq 0$ và $g_k=0$, thì sự khác biệt của các phần nguyên và phần thập phân của độ trễ tần số cơ bản được kiểm tra, sao cho:

○ Nếu phần nguyên và phần thập phân của độ trễ tần số cơ bản trong khung thứ k hiện tại và trong khung thứ $(k-1)$ trước đó là như nhau, thì:

- không có bước lọc thứ nhất cũng như bước lọc thứ hai;
- có bước lọc thứ ba dọc theo 100% khoảng thời gian cập nhật hiện tại;

○ nếu không thì, nếu có sự khác biệt về phần nguyên hoặc trong phần thập phân của độ trễ tần số cơ bản:

- có bước lọc thứ nhất trong ít nhất khoảng thời gian phụ ban đầu;
- có bước lọc thứ hai 52 trong ít nhất khoảng thời gian phụ ban đầu;
- có bước lọc thứ ba 53 trong khoảng thời gian phụ tiếp theo.

Theo các ví dụ, hệ thống bao gồm phía bộ mã hóa và phía bộ giải mã, trong đó ít nhất một trong các bộ lọc thứ nhất, thứ hai, thứ ba và/hoặc thứ tư nằm tại phía bộ giải mã.

Theo các ví dụ, hệ thống bao gồm phía bộ mã hóa và phía bộ giải mã, trong đó ít nhất một trong các bộ lọc thứ nhất, thứ hai, thứ ba và/hoặc thứ tư nằm tại phía bộ mã hóa.

Theo các ví dụ, hệ thống bao gồm bộ chuyển đổi để chuyển đổi phép biểu diễn thứ nhất của tín hiệu thông tin thành phép biểu diễn thứ hai của tín hiệu thông tin.

Theo các ví dụ, hệ thống được tạo cấu hình để:

xác định xem nếu bộ lọc thứ nhất và/hoặc thứ hai sẽ hoạt động như bộ lọc nhận dạng; và trong trường hợp xác định như vậy, thì bỏ qua bộ lọc thứ nhất và/hoặc thứ hai.

Theo các ví dụ, sáng chế đề xuất phương pháp lọc tín hiệu thông tin đầu vào, bao gồm các khoảng thời gian cập nhật khác nhau, theo các tham số tương ứng với các khoảng thời gian cập nhật, để thu được tín hiệu đầu ra được lọc, phương pháp bao gồm:

thực hiện bước lọc thứ nhất ít nhất tại khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại theo các tham số liên quan đến các khoảng thời gian cập nhật trước đó, trong đó các tham số dọc theo ít nhất là khoảng thời gian phụ ban đầu được thay đổi từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn; và

thực hiện bước lọc thứ hai ít nhất tại khoảng thời gian phụ ban đầu, theo các tham số liên quan đến các khoảng thời gian cập nhật hiện tại, trong đó các tham số dọc theo khoảng thời gian phụ ban đầu được thay đổi từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn; và

trong đó bước lọc thứ nhất được thực hiện trên tín hiệu thông tin đầu vào và bước lọc thứ hai được thực hiện trên tín hiệu thu được từ bước lọc thứ nhất.

Theo các ví dụ, sáng chế đề xuất bộ lưu trữ không tạm thời lưu trữ các lệnh, khi được chạy bởi bộ xử lý, khiến bộ xử lý thực hiện một trong các phương pháp nêu trên hoặc dưới đây và/hoặc để thực hiện hệ thống như nêu trên hoặc dưới đây và/hoặc thành phần của hệ thống đó.

Tín hiệu thông tin đầu vào có thể là, ví dụ, là tín hiệu âm thanh.

Trong một số ví dụ, khoảng thời gian là toàn bộ khung. Trong các ví dụ khác, khoảng thời gian là nhỏ hơn khung.

Theo đó, kỹ thuật trên chỉ có thể được thực hiện trong khoảng thời gian ban đầu hoặc khoảng thời gian cuối cùng của khung được xác định: trong khoảng thời gian tiếp theo, các tham số cho khung được xác định có thể được sử dụng, làm giảm thêm độ phức tạp tính toán.

Theo đó, không có sửa đổi đầu ra trong khoảng thời gian tiếp theo.

Trong các ví dụ, phép nội suy các hệ số có thể được sử dụng thay cho kỹ thuật được thảo luận nêu trên. Điều này có thể được kiểm soát trên cơ sở lựa chọn, để thích ứng tốt hơn các hoạt động lọc với các điều kiện tín hiệu.

Trong các ví dụ, có thể đặt ngưỡng thứ hai tại mức tối thiểu giữa phần nguyên của độ trễ tần số cơ bản tại khung được xác định và phần nguyên của độ trễ tần số cơ bản tại khung trước (hoặc khung tiếp theo) và/hoặc để đặt ngưỡng thứ hai đến mức tối đa giữa

độ khuếch đại tại khung được xác định và độ khuếch đại tại khung trước hoặc khung tiếp theo. Theo đó, có thể sử dụng bộ lọc thứ tư khi khoảng cách giữa độ khuếch đại tại khung được xác định và độ khuếch đại tại khung trước đó nhỏ hơn ngưỡng thứ hai và/hoặc sử dụng bộ lọc thứ nhất và thứ hai khi khoảng cách giữa phần nguyên của độ trễ tần số cơ bản tại khung được xác định và phần nguyên của độ trễ tần số cơ bản tại khung trước đó hoặc khung tiếp theo nhỏ hơn ngưỡng thứ hai.

Trong các ví dụ, có thể xác định điều kiện liên quan đến độ khuếch đại của tín hiệu tại khung được xác định và tại khung trước đó hoặc khung tiếp theo. Theo đó, có thể sử dụng bộ lọc thứ tư khi có ít nhất một độ khuếch đại của tín hiệu tại khung được xác định và tại khung trước đó hoặc khung tiếp theo bằng không. Trong các ví dụ, có thể sử dụng bộ lọc thứ nhất và thứ hai khi cả hai độ khuếch đại của tín hiệu tại khung xác định và tại khung trước đó hoặc khung tiếp theo khác không.

Theo đó, các tham số liên quan đến khung trước hoặc khung tiếp theo và các tham số liên quan đến khung xác định được sửa đổi (ví dụ: từng mẫu) để thực hiện lọc duyên dáng nhằm tránh và/hoặc giảm sự gián đoạn giữa các khung.

Theo đó, các tín hiệu đầu vào của các bộ lọc thứ nhất và thứ hai có thể được làm mờ dễ dàng và mạch lạc.

Theo các ví dụ, bộ lọc thứ nhất được tạo cấu hình để cung cấp tín hiệu đầu ra bộ lọc thứ nhất ở dạng

$$\widehat{x_{ltpf}}'(n) = \hat{x}(n) - \left(1 - \frac{n}{\frac{N_F}{4}}\right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x_{ltpf}}' \left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k\right) \right]$$

for $n = 0 \dots \frac{N_F}{4}$

và bộ lọc thứ hai được tạo cấu hình để cung cấp tín hiệu đầu ra đã lọc ở dạng

$$\widehat{x_{ltpf}}(n) = \widehat{x_{ltpf}}'(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x_{ltpf}}'(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}}' \left(n - p_{int} + \frac{L_{den}}{2} - k\right) \right]$$

for $n = 0 \dots \frac{N_F}{4}$

trong đó $\frac{N_F}{4}$ là độ dài của khoảng thời gian phụ ban đầu, $\hat{x}(n)$ là tín hiệu thông tin đầu vào, $\widehat{x_{ltpf}}'$ là tín hiệu trung gian, $\widehat{x_{ltpf}}(n)$ là tín hiệu đầu ra đã lọc, n là tức

thời, p_{int}^{mem} và p_{fr}^{mem} là tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khung trước đó, p_{int} và p_{fr} là tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khung xác định, $c_{num}(k)$ và $c_{den}(k, p_{fr})$ là các hệ số dựa trên giá trị độ khuếch đại cho khung được xác định, $c_{num}^{mem}(k)$ và $c_{den}(k, p_{fr})$ là các hệ số dựa trên giá trị độ khuếch đại cho khung trước đó, L_{den} and L_{num} được cố định và/hoặc dựa trên tốc độ lấy mẫu của tín hiệu đầu vào.

Trong các ví dụ, tại trạng thái lọc thấp hơn, các tham số của cùng một bộ lọc được sửa đổi để có giá trị gần 0 so với tại trạng thái lọc cao hơn. Ví dụ, các tham số có thể được định tỉ lệ theo các hệ số tỉ lệ thay đổi dần.

Sáng chế có thể được sử dụng, ví dụ, cho lọc sau dài hạn (Long Term Post Filtering - LTPF). Nó là công cụ để mã hóa âm thanh dựa trên biến đổi giúp giảm nhiễu liên âm điều hòa. Điều này dựa trên việc lọc sau mà được áp dụng lên tín hiệu miền thời gian sau khi giải mã biến đổi. Bộ lọc sau này về cơ bản là bộ lọc đáp ứng xung vô hạn (IIR) với đáp ứng tần số kiểu lược được điều khiển bởi một tham số (độ trễ tần số cơ bản) hoặc hai tham số (độ trễ tần số cơ bản và độ khuếch đại). Để có độ bền tốt hơn, các tham số sau lọc (độ trễ tần số cơ bản và độ khuếch đại trên mỗi khung) được ước tính tại phía bộ mã hóa và được mã hóa theo dòng bit khi độ khuếch đại khác không.

Mô tả vắn tắt các hình vẽ

Các hình vẽ Fig. 1, 1a, 2, 3, 4, 5a và 5b thể hiện các hệ thống theo các ví dụ;

Các hình vẽ Fig. 6, 7 và 7a thể hiện các phương pháp theo các ví dụ;

Các hình vẽ Fig. 8 và 9 thể hiện các hệ thống theo các ví dụ;

Các hình vẽ Fig. 10 và 11 thể hiện các biểu đồ theo thời gian theo các ví dụ;

Các hình vẽ Fig. 12a và 12b thể hiện các hệ thống theo các ví dụ; và

Fig. 13 thể hiện hàm biến đổi theo ví dụ

Mô tả chi tiết sáng chế

Theo cách tiếp cận được trình bày tại đây, các tham số lọc (còn được gọi là tham số hoặc tham số bộ lọc) nói chung khác nhau cho các khoảng thời gian cập nhật khác nhau. Hai khoảng thời gian cập nhật liên tiếp có thể có các tham số khác nhau. Theo các

ví dụ, khoảng thời gian cập nhật có thể là tín hiệu thích ứng và độ dài của nó có thể được thay đổi hoặc chuyển đổi theo thời gian.

Trong một số ví dụ, tín hiệu được chia thành các khung. Ví dụ: khung có thể liên quan đến một số lượng mẫu cố định và/hoặc độ dài thời gian cố định (ví dụ: 20 ms). Khi được truyền hoặc lưu trữ, khung có thể liên quan đến với các tham số cụ thể, ví dụ: các tham số lọc. Trong cùng một khung, các tham số có thể là hằng số chung.

Trong một số ví dụ, khoảng thời gian cập nhật có thể tương ứng với khung. Do đó, khi được truyền hoặc lưu trữ, tín hiệu có thể chứa các tham số (ví dụ: các tham số lọc) liên quan đến khung (cũng là khoảng thời gian cập nhật), ví dụ: bằng dữ liệu được mã hóa liên quan đến khung cụ thể.

Trong một số ví dụ, khoảng thời gian cập nhật không tương ứng với khung định trước. Khi được truyền hoặc lưu trữ, tín hiệu có thể bao gồm các tham số liên quan đến khoảng thời gian cập nhật ngay cả khi chúng được báo hiệu là được liên kết với khung. Trong một số trường hợp, khung mới (với các tham số mới) đoán trước khoảng thời gian cập nhật mới và khoảng thời gian cập nhật cũ vẫn được liên kết với các tham số cũ. Khi các khoảng thời gian cập nhật là thích ứng (ví dụ: độ dài của chúng được xác định một cách nhanh chóng, trên cơ sở các đặc tính của tín hiệu, chẳng hạn), tất nhiên chúng nói chung không tương ứng với các khung có độ dài cố định. Do đó, có thể liên kết các tham số với một khoảng thời gian cập nhật cụ thể (trong một số trường hợp, được liên kết với một khung).

Fig. 1 thể hiện hệ thống 10 có thể là hệ thống lọc tín hiệu đầu vào 11 (được biểu thị như là "x" hay "x [n]", trong đó, "n" đề cập đến mẫu cụ thể).

Tín hiệu 11 có thể là tín hiệu thông tin, chẳng hạn như tín hiệu âm thanh. Phép biểu diễn dạng số của tín hiệu thông tin có thể được sử dụng. Tín hiệu có thể bao gồm một chuỗi các mẫu, mỗi mẫu thu được tại một thời điểm khác nhau (ví dụ: các thời điểm riêng biệt). Tín hiệu có thể được chia thành các khung khác nhau và/hoặc khoảng thời gian cập nhật (ví dụ: một chuỗi các mẫu). Mỗi khung và/hoặc khoảng thời gian cập nhật có thể được cấu thành bởi nhiều mẫu (ví dụ: 1, 2, ..., n,...), ví dụ, mỗi khung được liên kết với một thời điểm riêng biệt). Mỗi khung và/hoặc khoảng thời gian cập nhật có thể

được chia thành khoảng thời gian phụ ban đầu và khoảng thời gian phụ tiếp theo (khoảng thời gian phụ có thể là khoảng thời gian phụ thích hợp, vì độ dài của nó có thể nhỏ hơn độ dài của khoảng thời gian cập nhật). Nói chung, các mẫu của khoảng thời gian phụ ban đầu đứng trước (là trước) các mẫu của khoảng thời gian phụ tiếp theo của cùng một khung và/hoặc khoảng thời gian cập nhật. Khung được xác định (hiện tại, được xử lý hiện tại) đứng trước khung tiếp theo và/hoặc khoảng thời gian cập nhật và được đặt trước khung trước (khung trước, cũ). Khoảng thời gian cập nhật được xác định (hiện tại, hiện thời) đứng trước khoảng thời gian cập nhật tiếp theo và được đứng trước bởi khoảng thời gian cập nhật trước đó (trước, cũ). Khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật có thể có độ dài thời gian nằm trong khoảng từ 1% đến 99%, cụ thể hơn là 20% và 30% (ví dụ: một phần tư), của độ dài thời gian của khoảng thời gian cập nhật hiện tại. Khoảng thời gian phụ tiếp theo có thể có độ dài thời gian nằm trong khoảng từ 1% đến 99%, cụ thể hơn là 70% và 80% (ví dụ: ba phần tư), của độ dài thời gian của khoảng thời gian cập nhật hiện tại.

Với một số ví dụ, nó được gọi là ít nhất một khoảng thời gian phụ ban đầu" của khoảng thời gian cập nhật, ngụ ý rằng 100% khoảng thời gian cập nhật có thể được đề cập trong một số ví dụ.

Hệ thống 10 có thể lọc tín hiệu thông tin đầu vào 11 (x) theo các tham số thay đổi theo các khoảng thời gian cập nhật (ví dụ: các tham số mà, nói chung, thay đổi theo thời gian theo khoảng thời gian cập nhật cụ thể mà chúng được liên kết, ví dụ, tùy theo các tham số được mã hóa và được liên kết với khung cụ thể). Hệ thống 10 có thể cung cấp tín hiệu đầu ra được lọc 15 (y hoặc y [n]). Hệ thống 10 có thể cung cấp tín hiệu âm thanh được lọc sau (ví dụ: miền thời gian (time domain - TD, biểu diễn tín hiệu âm thanh được lọc sau).

Hệ thống 10 có thể bao gồm bộ lọc thứ nhất 12 để lọc tín hiệu thông tin đầu vào 11 (tín hiệu đầu vào bộ lọc thứ nhất). Bộ lọc thứ nhất 12 có thể hoạt động với các mẫu có ít nhất khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại (khoảng thời gian cập nhật hiện tại, chẳng hạn như khoảng thời gian cập nhật hiện tại đang được xử lý), để thu được tín hiệu đầu ra bộ lọc thứ nhất là tín hiệu trung gian 13 (y 'hoặc y' [n]). (Ít nhất là khoảng thời gian phụ ban đầu có thể đề cập đến, ví dụ, chỉ trong các khoảng

thời gian phụ ban đầu hoặc, trong các ví dụ, đến khoảng thời gian lớn hơn khoảng thời gian phụ ban đầu, chẳng hạn như khoảng thời gian cập nhật). Tín hiệu trung gian 13 có thể thu được bằng cách sử dụng các tham số liên quan đến khoảng thời gian cập nhật trước đó (ví dụ: khoảng thời gian cập nhật trước khoảng thời gian cập nhật hiện tại). Bộ lọc thứ nhất 12 được tạo cấu hình để thay đổi (ví dụ: dần dần, ví dụ, bằng cách làm giảm dần, ví dụ, một cách đơn điệu) các tham số dọc theo ít nhất là khoảng thời gian phụ ban đầu từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn. Ví dụ, các tham số có thể được giảm bớt và/hoặc làm mờ bớt (do đó ngụ ý trạng thái lọc cao hơn) tương ứng với các mẫu thứ nhất trong khoảng thời gian phụ ban đầu. Các tham số có thể được giảm hơn nữa và/hoặc được làm mờ hơn nữa (do đó ngụ ý trạng thái lọc thấp hơn) tương ứng với các mẫu cuối cùng trong ít nhất là khoảng thời gian phụ ban đầu (trong đó các tham số có thể được xử lý để gần 0 hơn).

Hệ thống 10 có thể bao gồm bộ lọc thứ hai 14. Bộ lọc thứ hai 14 có thể có tín hiệu đầu vào bộ lọc thứ hai và tín hiệu đầu ra bộ lọc thứ hai. Bộ lọc thứ hai 14 có thể lọc tín hiệu trung gian 13 (là tín hiệu đầu vào bộ lọc thứ hai). Bộ lọc thứ hai 14 có thể hoạt động với các mẫu ít nhất là khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại (ví dụ: cùng ít nhất khoảng thời gian phụ ban đầu mà bộ lọc 12 hoạt động trên đó). Bộ lọc thứ hai 14 có thể lọc tín hiệu trung gian theo các tham số liên quan đến khoảng thời gian cập nhật hiện tại. Bộ lọc thứ hai 14 được tạo cấu hình để thay đổi (ví dụ: dần dần, ví dụ, bằng cách làm giảm dần, ví dụ, theo cách đơn điệu) các tham số dọc theo ít nhất là khoảng thời gian phụ ban đầu từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn. Ví dụ, các tham số có thể được giảm hơn nữa và/hoặc được làm mờ hơn nữa (do đó ngụ ý trạng thái lọc thấp hơn) cho các mẫu thứ nhất trong ít nhất là khoảng thời gian phụ ban đầu (trong đó các tham số có thể được xử lý để là 0 hoặc gần đến 0). Các tham số có thể được giảm bớt và/hoặc làm mờ bớt (do đó có nghĩa là trạng thái lọc cao hơn) với các mẫu cuối cùng trong ít nhất khoảng thời gian phụ ban đầu.

Trạng thái lọc thấp hơn có thể sao cho đáp ứng xung gần với đáp ứng xung của bộ lọc nhận dạng hơn so với đáp ứng xung trong trạng thái lọc cao hơn. Trong một số ví dụ, trạng thái lọc thấp hơn và trạng thái lọc cao hơn có thể sao cho trạng thái lọc thấp hơn có nghĩa là tăng năng lượng của đáp ứng xung so với năng lượng của đáp ứng xung

của trạng thái lọc cao hơn. Trong một số ví dụ, trạng thái lọc thấp hơn và trạng thái lọc cao hơn có thể sao cho trạng thái lọc thấp hơn có nghĩa là năng lượng đã giảm của đáp ứng xung so với năng lượng của đáp ứng xung của trạng thái lọc cao hơn. Khi bộ lọc thứ nhất 12 hoạt động tại trạng thái lọc thấp hơn, có nghĩa là đáp ứng xung khác với (ví dụ: thấp hơn hoặc cao hơn) năng lượng có được bởi cùng một bộ lọc thứ nhất 12 khi nó hoạt động tại trạng thái lọc cao hơn. Tương tự áp dụng cho bộ lọc thứ hai 14. Trong trạng thái lọc thấp hơn, các tham số có thể gần 0 hơn so với tại trạng thái lọc cao hơn.

Ví dụ: đơn vị lọc thứ nhất 12 có thể thay đổi từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn và/hoặc đơn vị lọc thứ hai 14 có thể thay đổi từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn.

Nói chung, khi thay đổi từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn, các tham số có thể được sửa đổi (ví dụ: dần dần và/hoặc đơn điệu và/hoặc bằng cách giảm và/hoặc bằng cách làm mờ). Trong trạng thái lọc thấp hơn, các tham số có thể được xử lý là 0 hoặc có thể được xử lý gần 0 hơn các tham số tại trạng thái lọc cao hơn. Ví dụ: tại trạng thái lọc cao hơn, các tham số có thể được giảm ít hơn và/hoặc được làm mờ ít hơn và/hoặc xa hơn 0 so với trạng thái lọc thấp hơn.

Trong trạng thái lọc thấp hơn, hiệu quả lọc có thể bị giảm (ví dụ: hiệu quả tương tự của hoặc hiệu quả gần với hiệu quả của bộ lọc nhận dạng có thể thu được). Tại trạng thái lọc cao hơn, có thể thu được hiệu ứng lọc mạnh.

Tại trạng thái lọc cao hơn, tín hiệu đầu vào được lọc mạnh. Tại trạng thái lọc thấp hơn, tín hiệu đầu vào được lọc mạnh. Trong các ví dụ, các tham số được làm mờ càng nhiều, trạng thái lọc càng thấp.

Các bộ lọc thứ nhất và/hoặc thứ hai 12, 14 có thể được tạo cấu hình để làm mờ và/hoặc giảm các tham số sao cho tại trạng thái lọc cao hơn, các tham số được giảm và/hoặc được làm mờ được làm mờ (ví dụ: xa 0 hơn) và/hoặc được giảm ít hơn tại trạng thái lọc thấp hơn (ví dụ: gần 0).

Bộ lọc thứ nhất 12 có thể được tạo cấu hình để định tỉ lệ (ví dụ: hạ thấp) các tham số bằng cách sử dụng hệ số làm mờ thứ nhất hoặc hệ số tỉ lệ và/hoặc bộ lọc thứ hai 14 có thể được tạo cấu hình để định tỉ lệ (ví dụ: hạ thấp) các tham số sử dụng hệ số làm mờ

thứ hai hoặc hệ số tỉ lệ. Hệ số làm mờ thứ nhất hoặc hệ số tỉ lệ và/hoặc hệ số làm mờ thứ hai hoặc hệ số tỉ lệ có thể gần 0 tại trạng thái lọc thấp hơn so với trạng thái lọc cao hơn.

Bộ lọc thứ nhất và/hoặc thứ hai 12, 14 có thể được tạo cấu hình để làm mờ và/hoặc giảm và/hoặc thu nhỏ các tham số để các tham số được làm mờ và/hoặc giảm bởi các hệ số làm mờ thứ nhất và/hoặc các hệ số tỉ lệ gần 0 tại trạng thái lọc thấp hơn so với trạng thái lọc cao hơn.

Bộ lọc thứ nhất 12 có thể được tạo cấu hình để thay đổi các tham số từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn bằng cách làm mờ dần và/hoặc giảm các tham số liên quan đến khoảng thời gian cập nhật trước đó và/hoặc bộ lọc thứ hai 14 có thể được tạo cấu hình để thay đổi các tham số bằng cách tăng dần và/hoặc phóng to từ các tham số gần 0 đến các tham số liên kết với khoảng thời gian cập nhật hiện tại.

Bộ lọc thứ nhất 12 có thể được tạo cấu hình để thay đổi các tham số từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn bằng cách sử dụng việc giảm các hệ số làm mờ và/hoặc các hệ số tỉ lệ (ví dụ: bằng cách sử dụng các hệ số làm mờ và/hoặc các hệ số tỉ lệ, ví dụ dần dần, đi về phía 0). Bộ lọc thứ hai 14 có thể được tạo cấu hình để thay đổi (ví dụ, dần dần) các tham số từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn bằng cách mở rộng (ví dụ, dần dần) các hệ số làm mờ và/hoặc các hệ số tỉ lệ (ví dụ: bằng cách sử dụng các hệ số làm mờ và/hoặc các hệ số tỉ lệ, ví dụ dần dần, đi xa khỏi 0).

Bộ lọc thứ nhất 12 có thể được tạo cấu hình để thay đổi các tham số từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn bằng cách áp dụng hàm tạo cửa sổ giảm cho các tham số (ví dụ: hàm tạo cửa sổ đi về 0, ví dụ dần dần) và/hoặc bộ lọc thứ hai 14 có thể được tạo cấu hình để thay đổi các tham số từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn bằng cách áp dụng hàm tạo cửa sổ mở rộng cho các tham số (ví dụ: hàm tạo cửa sổ đi xa khỏi 0, ví dụ dần dần).

Các tham số có thể được cung cấp, ví dụ, như dữ liệu điều khiển trong tín hiệu đầu vào 11. Ví dụ, các bộ lọc có thể là bộ lọc đáp ứng xung vô hạn bất biến theo thời gian tuyến tính (Liner Time Invariant - LTI) (IIR) (ví dụ: đối với lọc sau LTP). Các tham số có thể là hoặc bao gồm, ví dụ, thông tin độ khuếch đại và/hoặc thông tin tần số cơ bản.

Cụ thể, bộ lọc thứ nhất và thứ hai 12 và 14 có thể là một phần của bộ lọc LTP (hoặc bộ lọc sau), ví dụ: tại bộ giải mã (ví dụ: bộ giải mã âm thanh). Ví dụ, các tham số có thể thu được từ các phép đo dựa trên sự điều hòa. Ví dụ: các tham số có thể dựa trên độ trễ tần số cơ bản T , trên phần nguyên T_{int} của độ trễ tần số cơ bản, trên phần thập phân T_{fr} của độ trễ tần số cơ bản và/hoặc trên độ khuếch đại g , ví dụ, thu được tại bộ giải mã âm thanh. Các tham số có thể được liên kết, ví dụ, với khoảng thời gian cập nhật (trong một số ví dụ là khung có độ dài cố định hoặc có thể có độ dài thích ứng).

Do đó, mỗi bộ lọc thứ nhất và thứ hai có thể được liên kết với các tham số cụ thể và/hoặc với hàm chuyển cụ thể. Cụ thể, hàm chuyển có thể là loại có tử số và mẫu số, trong đó tử số bao gồm giá trị độ khuếch đại được biểu thị bởi thông tin độ khuếch đại, và trong đó mẫu số bao gồm phần nguyên của độ trễ tần số cơ bản được biểu thị bởi thông tin độ trễ tần số cơ bản và bộ lọc đa chạm phụ thuộc vào phần thập phân của độ trễ tần số cơ bản. Ví dụ, hàm chuyển có thể là:

$$H(z) = \frac{1 - \alpha\beta g B(z, 0)}{1 - \beta g B(z, T_{fr}) z^{-T_{int}}}$$

các tham số của nó có thể được xác định từ các tham số được ước lượng tại phía bộ mã hóa và/hoặc được giải mã từ dòng bit. g có thể là độ khuếch đại được giải mã, T_{int} và T_{fr} là phần nguyên và phần thập phân của độ trễ tần số cơ bản được giải mã, α và β hai vô hướng mà gán trọng số độ khuếch đại, và $B(z, T_{fr})$ là bộ lọc FIR mà các hệ số của nó phụ thuộc vào phần thập phân của độ trễ tần số cơ bản được giải mã. Thứ tự và các hệ số của $B(z, T_{fr})$ có thể cũng phụ thuộc vào tốc độ truyền dữ liệu và tốc độ lấy mẫu đầu ra. Đáp ứng tần số khác nhau có thể được thiết kế và tinh chỉnh cho mỗi tổ hợp của tốc độ truyền dữ liệu và tốc độ lấy mẫu đầu ra. Ví dụ về hàm chuyển 130 được cung cấp trên Fig. 13 (các loại bộ lọc và/hoặc hàm chuyển khác mặc dù có thể có).

Đáng chú ý, các tham số và hàm chuyển có thể thay đổi cho từng khoảng thời gian cập nhật (mà có thể là một khung theo đó tín hiệu gốc có thể được chia nhỏ). Do đó, khoảng thời gian cập nhật thứ k có thể liên quan đến hàm chuyển $H_k(z)$ và các tham số như $T_{int,k}$, $T_{fr,k}$, g_k , trong khi khoảng thời gian cập nhật thứ $(k-1)$ có thể liên quan đến hàm chuyển $H_{k-1}(z)$ và các tham số như $T_{int,k-1}$, $T_{fr,k-1}$, g_{k-1} . Do đó tại khung thứ k hoặc khoảng thời gian cập nhật, bộ lọc thứ nhất 12 có thể hoạt động sử dụng các tham số cũ

$T_{int,k-1}$, $T_{fr,k-1}$, g_{k-1} , trong khi bộ lọc thứ hai 14 có thể hoạt động sử dụng các tham số cập nhật $T_{int,k}$, $T_{fr,k}$, g_k . Quy trình này có thể được thực hiện cho ít nhất khoảng thời gian phụ ban đầu (hoặc, theo một số ví dụ, với 100%) của khoảng thời gian cập nhật thứ k.

Nói chung, mỗi hoặc ít nhất một trong các bộ lọc tại các phần tử 10 và 12 có thể là bộ lọc LTI IIR (cũng có thể được biểu thị là $H(z)$) được biểu thị dưới dạng:

$$H(z) = \frac{1 + \sum_{i=0}^P b_i z^{-i}}{1 + \sum_{j=1}^Q a_j z^{-j}}$$

hoặc sử dụng phương trình chênh lệch tuyến tính

$$y[n] = x[n] + \sum_{i=0}^P b_i x[n-i] - \sum_{j=1}^Q a_j y[n-j]$$

các hệ số b_i và a_j có thể là các tham số lọc. Đáng chú ý, các hệ số b_i và a_j có thể thay đổi, nói chung, với các khung khác nhau và/hoặc các khoảng thời gian cập nhật khác nhau.

Cần lưu ý rằng tín hiệu được lọc 15 thu được khi chuyển tiếp suôn sẻ giữa khoảng thời gian cập nhật thứ (k-1) trước đó và khoảng thời gian cập nhật thứ k hiện tại. Do đó, sự gián đoạn giữa các khoảng cập nhật khác nhau được tránh và/hoặc giảm.

Hơn nữa, các quy trình để thực hiện các chức năng lọc có độ phức tạp giảm đặc biệt.

Loại hệ thống này có thể được sử dụng, ví dụ, cho lọc sau dài hạn (Long Term Post Filtering - LTPF).

Các bộ lọc thứ nhất và thứ hai 12 và 14 có thể được coi là được nối theo chuỗi (hoặc xếp tầng, hoặc một bộ lọc này sau bộ lọc khác).

Fig. 1a thể hiện biến thể 10' của hệ thống 10 trong đó bộ lọc thứ nhất 12 và bộ lọc thứ hai 14 lần lượt có thể được bỏ qua tại các bộ chọn 16 và 17. Các bộ chọn 16 và 17 có thể được điều khiển, ví dụ, bằng các tham số (trên Fig. 1a, có thể thấy rằng điều kiện bên ngoài có thể gây ra việc bỏ qua của một bộ lọc).

Fig. 2 thể hiện hệ thống 10 có thể thực hiện hệ thống của Fig. 1 (cũng có thể là biến thể 10', ngay cả khi các kết nối bỏ qua của Fig. 1a tại đây không được thể hiện trên

Fig. 2 vì mục đích rõ ràng). Thông tin liên quan đến khoảng thời gian cập nhật thứ k hiện tại của tín hiệu thông tin đầu vào (tín hiệu đầu vào bộ lọc thứ nhất) 11 có thể bao gồm phép biểu diễn tín hiệu 11a (ví dụ: giá trị thực của các mẫu trong các khoảng khắc khác nhau trong miền thời gian mà tạo thành dữ liệu "x") và dữ liệu điều khiển 11b (ví dụ, có thể được mã hóa theo dòng bit và được truyền từ bộ phát hoặc được lưu trong bộ nhớ). Dữ liệu điều khiển 11b có thể bao gồm các tham số liên quan đến việc lọc tại khung thứ k hoặc khoảng thời gian cập nhật (ví dụ: độ trễ tần số cơ bản được ước lượng và/hoặc các giá trị độ khuếch đại hoặc các giá trị tương tự hoặc các phiên bản được xử lý của nó). Các tham số cho khung thứ k hiện tại hoặc khoảng thời gian cập nhật có thể được lưu trữ trong phần tử lưu trữ tham số 21a (ví dụ: bộ nhớ, chẳng hạn như thanh ghi). Phần tử lưu trữ tham số 21b có thể chứa các tham số của khung thứ $(k-1)$ trước đó hoặc khoảng thời gian cập nhật. Mũi tên 21 'đề cập đến thực tế là các tham số cho khung thứ $(k-1)$ hoặc khoảng thời gian cập nhật trước đó (được lưu trữ trước đó tại phần tử lưu trữ 21a) trở thành các tham số của khung hoặc khoảng thời gian cập nhật trước đó và được lưu trữ tại bộ lưu trữ phần tử 21b khi khung thứ k hoặc khoảng thời gian cập nhật mới được xử lý.

Các tham số được lưu trữ trong phần tử lưu trữ 21b (đã được chuẩn bị cho khung thứ $(k-1)$ trước đó) có thể được áp dụng cho phần lọc thứ nhất 22 (mà có thể thực hiện các hàm của bộ lọc thứ nhất 12) cho ít nhất là khoảng thời gian phụ ban đầu của khung thứ k hoặc khoảng thời gian cập nhật hiện tại. Các tham số được lưu trữ trong phần tử lưu trữ 21a (đã được chuẩn bị cho khung thứ k hiện tại) có thể được áp dụng cho phần lọc thứ hai 24 (mà có thể thực hiện các hàm của bộ lọc thứ hai 14) cho ít nhất là khoảng thời gian phụ ban đầu của khung thứ k hoặc khoảng thời gian cập nhật hiện tại.

Tuy nhiên, các tham số được lưu trữ trong phần tử lưu trữ 21b (được chuẩn bị ban đầu cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó) có thể bị thay đổi, ví dụ, bởi khối 23b, khiến phần lọc thứ nhất 22 di chuyển (ví dụ: dần dần, ví dụ, đơn điệu) từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn trong khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại. Ví dụ: các tham số có thể được định tỉ lệ (ví dụ: thu nhỏ, làm mờ) bằng cách sử dụng hệ số làm mờ về 0, ví dụ: hệ số làm mờ giảm (ví dụ: làm mờ bằng cách nhân). Khối 23b có thể áp dụng hàm tạo cửa sổ. Hàm tạo cửa sổ có

thể thay đổi về 0 và/hoặc giảm ít nhất là khoảng thời gian phụ ban đầu (ví dụ: từ giá trị dương cao nhất, ví dụ 1, tại mẫu thứ nhất của khoảng cập nhật hiện tại thứ k, thành giá trị thấp nhất, ví dụ 0, tại mẫu cuối cùng của ít nhất thời gian phụ ban đầu).

Ví dụ: có thể thu được hiệu quả lọc giảm (ví dụ: từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn) bằng cách giảm dần hệ số làm mờ (ví dụ: từ giá trị tối đa, ví dụ 1, xuống giá trị tối thiểu hoặc không đáng kể, ví dụ 0). Khi hệ số làm mờ không đáng kể (hoặc 0), các tham số được sửa đổi thành giá trị không đáng kể (hoặc 0) và đầu ra 13 (y') của phần lọc thứ nhất 22 gần như tương tự (hoặc giống với) tín hiệu thông tin đầu vào 11 (x).

Các tham số được lưu trữ trong phần tử lưu trữ 21a (được liên kết với khung thứ k hiện tại hoặc khoảng thời gian cập nhật hiện tại) có thể được thay đổi bởi khối 23a để khiến phần lọc thứ hai 24 di chuyển (ví dụ: dần dần, ví dụ, đơn điệu) từ trạng thái lọc thấp hơn đến trạng thái lọc cao hơn trong ít nhất là khoảng thời gian phụ ban đầu của khung thứ k hoặc khoảng thời gian cập nhật hiện tại. Ví dụ: các tham số có thể được định tỉ lệ (ví dụ: thu nhỏ, làm mờ) bằng cách tăng hệ số làm mờ (ví dụ: bằng cách nhân), có thể mở rộng từ 0 hoặc giá trị gần 0 đến giá trị xa 0 hơn. Khối 23b có thể áp dụng hàm tạo cửa sổ. Hàm tạo cửa sổ có thể tăng hoặc thay đổi (ví dụ: từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn) từ thời điểm ban đầu của khoảng thời gian phụ ban đầu sang thời điểm cuối cùng của khoảng thời gian phụ ban đầu (ví dụ: từ giá trị gần với 0 đến một giá trị xa 0 hơn và/hoặc từ một giá trị thấp nhất tại mẫu đầu tiên của khoảng thời gian phụ ban đầu đến giá trị cao nhất dương, ví dụ 1, tại mẫu cuối cùng của khoảng thời gian phụ ban đầu của khung thứ k hiện tại hoặc khoảng thời gian cập nhật hiện tại hoặc tại mẫu cuối cùng của khung hoặc khoảng thời gian cập nhật).

Ví dụ: có thể tăng hiệu quả lọc (ví dụ: từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn) bằng cách mở rộng dần từ 0 hoặc giá trị gần 0 đến giá trị xa 0 hơn, ví dụ: mở rộng hoặc tăng (ví dụ, đơn điệu hoặc hoàn toàn đơn điệu) hệ số làm mờ (ví dụ: từ giá trị gần 0 đến giá trị khác xa 0 hơn và/hoặc từ giá trị tối thiểu hoặc không đáng kể, ví dụ 0, đến giá trị tối đa, ví dụ: 1). Khi hệ số làm mờ không đáng kể (hoặc 0), các tham số được sửa đổi thành giá trị không đáng kể (hoặc 0) và đầu ra của phần lọc thứ hai 24 gần như tương tự (hoặc giống với) tín hiệu đầu vào của nó (mà là tín hiệu trung gian y' hoặc 13).

Trong các ví dụ, các tham số cho các bộ lọc thứ nhất và/hoặc thứ hai 12, 14 có thể được sửa đổi trong quá trình nối tiếp các mẫu của ít nhất là khoảng thời gian phụ ban đầu, theo các hệ số (ví dụ: các hệ số định tỉ lệ của các khối 23a, 23b) mà bổ sung với nhau đến một giá trị không đổi (ví dụ: giá trị dương, chẳng hạn như 1), sao cho tổng của chúng không đổi. Trong các ví dụ, sự thay đổi của các hệ số có thể là tuyến tính (ví dụ: có thể mô tả bằng phương trình bậc 1).

Trong các ví dụ, phần lọc thứ nhất 22 và phần lọc thứ hai 24 có thể có chung cấu trúc phần cứng/phần mềm, có đầu ra của chúng thay đổi chỉ nhờ vào đầu vào của các đầu vào và/hoặc các tham số và/hoặc các hệ số khác nhau.

Đáng chú ý, các tham số có thể được liên kết, trong các phần tử lưu trữ 21a và 21b, với một độ dài cụ thể của khoảng thời gian phụ hoặc với 100% khoảng thời gian cập nhật. Do đó, có thể biết rằng phần trăm (hoặc trong mọi trường hợp là phần) của khoảng thời gian cập nhật hoặc khung sẽ được làm nhẵn. Trong một số trường hợp, lựa chọn người dùng (ví dụ: được thiết lập trong suốt phiên cấu hình) có thể xác định độ dài của ít nhất khoảng thời gian phụ.

Fig. 3 thể hiện hệ thống 30 mà có thể bao gồm phần lọc 10 (hoặc 10', ngay cả khi các kết nối bỏ qua không được hiển thị trên Fig.3) cho ít nhất khoảng thời gian phụ ban đầu và phần lọc 31 cho khoảng thời gian phụ tiếp theo (trong đó kỹ thuật lọc kép của Fig. 1 và 2 được sử dụng trong 100% của khoảng thời gian cập nhật, "phần lọc của khoảng thời gian phụ tiếp theo" không hoạt động). Phần lọc cho khoảng thời gian phụ ban đầu có thể giống với hệ thống 10. Phần lọc 31 cho khoảng thời gian phụ tiếp theo có thể được tạo cấu hình để lọc tín hiệu trong khoảng thời gian cập nhật thứ k hiện tại sau khi kết thúc phần được lọc bởi phần lọc 10 cho ít nhất là khoảng thời gian phụ ban đầu (có thể là khoảng thời gian tiếp theo sau khoảng thời gian phụ ban đầu). Phần lọc 31 cho khoảng thời gian tiếp theo có thể là bộ lọc thứ ba thiếu các tham số làm mờ. Do đó, phần lọc thứ ba 31 có thể chỉ cần áp dụng các tham số được chuẩn bị ban đầu cho khoảng thời gian cập nhật hiện tại.

Bộ chọn 32 có thể theo dõi tín hiệu thông tin đầu vào 11 và thay đổi giữa việc sử dụng phần lọc 10 cho khoảng thời gian phụ ban đầu và sử dụng phần lọc 31 cho khoảng thời gian tiếp theo. Đáng chú ý, phần lọc 31 cho khoảng thời gian tiếp theo (bộ lọc thứ

ba) có thể được tạo từ các khối cấu trúc và/hoặc chức năng được sử dụng cho bộ lọc thứ nhất và/hoặc thứ hai 12 và 14.

Theo cách bổ sung hoặc thay thế, bộ chọn 32 có thể quyết định xem, trong suốt khoảng thời gian phụ ban đầu, phần lọc 10 (với bộ lọc thứ nhất và thứ hai 12, 14) sẽ được sử dụng cho khoảng thời gian phụ ban đầu hay nếu phần lọc 31 sẽ được sử dụng cho ít nhất là khoảng thời gian phụ ban đầu (bên cạnh việc sử dụng phần lọc 31 cho khoảng thời gian phụ tiếp theo, trong trường hợp). Quyết định có thể dựa trên các điều kiện cụ thể, ví dụ, có thể dựa trên các tham số (ví dụ: dựa trên so sánh giữa các giá trị tham số liên quan đến khoảng thời gian cập nhật hiện tại và giá trị của các tham số liên quan đến khoảng thời gian cập nhật trước đó). Một số ví dụ về quyết định này được cung cấp trong các đoạn sau.

Các bộ lọc tại các phần tử 22, 24, và 31 có thể là các bộ lọc sau LTP như đã thảo luận ở trên hoặc hơn nữa, nói chung là các bộ lọc LTI IIR (mà có thể cũng được biểu diễn như $H(z)$) và có thể được thể hiện ở dạng :

$$H(z) = \frac{1 + \sum_{i=0}^P b_i z^{-i}}{1 + \sum_{j=1}^Q a_j z^{-j}}$$

hoặc sử dụng phương trình chênh lệch tuyến tính

$$y[n] = x[n] + \sum_{i=0}^P b_i x[n-i] - \sum_{j=1}^Q a_j y[n-j]$$

các hệ số b_i và a_j có thể là các tham số lọc. Fig. 10 thể hiện khoảng thời gian cập nhật T liên kết với bộ lọc H_k (ví dụ: tại khoảng thời gian cập nhật thứ k hiện tại) và khoảng thời gian trước đó liên kết với bộ lọc H_{k-1} (ví dụ: tại khoảng ngày thứ $(k-1)$ trước đó). Chú ý là, Fig. 11 thể hiện khoảng thời gian cập nhật (hoặc khung) T , khoảng thời gian phụ ban đầu T_1 trong đó cả hai bộ lọc được sử dụng, và khoảng thời gian phụ tiếp theo $T_s = T - T_1$, trong số chỉ phần lọc 31 cho khoảng thời gian phụ tiếp theo được sử dụng. Trong các ví dụ trong đó 100% khoảng thời gian cập nhật (hoặc khung) T được lọc hai lần bởi các phần tử 12 và 14, có thể hiểu được rằng $T = T_1$, tức là khoảng thời gian phụ là giống với khoảng thời gian và khoảng thời gian tiếp theo không tồn tại)

Chúng ta xem xét đến bộ lọc thay đổi theo thời gian mà là trong khoảng thời gian cập nhật k bằng với bộ lọc LTI IIR H_k (mà là bộ sau LTP):

$$H_k(z) = \frac{1 + \sum_{i=0}^P b_{k,i} z^{-i}}{1 + \sum_{j=1}^Q a_{k,j} z^{-j}}$$

$$y[n] = x[n] + \sum_{i=0}^P b_{k,i} x[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j], kT \leq n < (k+1)T$$

Trong đó T liên quan đến khoảng thời gian cập nhật (và có thể bao gồm một số lượng mẫu riêng biệt) và k là chỉ số của nó, $k-1$ được liên kết với khoảng cập nhật thứ $(k-1)$ trước đó. Phần lọc thứ ba 31 có thể thuộc loại này. P và Q có thể là cụ thể cho bộ lọc (ví dụ: lần lượt là thứ tự bộ lọc tiếp theo và thứ tự bộ lọc phản hồi). Q có thể là ví dụ liên quan đến giá trị tối đa có thể cho T_{int} .

Các phần tử bộ lọc thứ nhất 12 và/hoặc 22 có thể xuất tín hiệu trung gian y' ở dạng:

$$y'[n] = x[n] + s_{k-1}[n] \left(\sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y'[n-j] \right), kT \leq n < kT + T_l$$

trong đó $s_{k-1}[n]$ thay đổi về 0 khi n tăng với $kT - T_l \leq n < (k+1)T$

ví dụ $s_{k-1}[n] = 1 - \frac{n-kT}{T_l}, kT \leq n < kT + T_l$

trong đó T_l liên quan đến khoảng thời gian phụ ban đầu.

Các phần tử bộ lọc thứ hai 14 và/hoặc 24 có thể xuất tín hiệu đầu ra đã lọc y ở dạng:

$$y[n] = y'[n] + s_k[n] \left(\sum_{i=0}^P b_{k,i} y'[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j] \right), kT \leq n < kT + T_l$$

$$s_k[n] = \frac{n-kT}{T_l} = 1 - s_{k-1}[n], kT \leq n < kT + T_l$$

Đáng chú ý, giá trị đầu ra được lọc $y[n]$ có đầu vào dựa trên đầu ra bộ lọc trung gian $y'[n]$ được cung cấp bởi bộ lọc thứ nhất.

Fig. 4 thể hiện hệ thống 40 như là một sửa đổi của hệ thống của Fig. 2. Bộ chọn (không thể hiện trên hình vẽ) có thể thay đổi từ việc sử dụng chế độ thứ nhất mà khoảng

thời gian phụ ban đầu (T_l) được lọc (như bởi hệ thống 10) và việc sử dụng chế độ thứ hai trong đó khoảng thời gian phụ tiếp theo của khoảng thời gian cập nhật hiện tại được lọc. Như được biểu diễn bởi bộ chuyển hướng 41a và công tắc 41b, sau khi đã lọc tín hiệu thông tin đầu vào x tại khoảng thời gian phụ ban đầu như trên Fig. 2, phần lọc thứ nhất 22 có thể được bỏ qua bởi bộ chuyển hướng 41a. Theo đó, phần lọc thứ hai 24 có thể được cấp trực tiếp với biểu diễn 11a của tín hiệu thông tin đầu vào x . Như được biểu diễn bởi bộ chuyển hướng 41c, các tham số được lưu trữ trong phần tử lưu trữ 21a có thể được sử dụng trực tiếp tại phần lọc thứ hai 24 mà không bị làm mờ, tức là bỏ qua khối 23a. Do đó, trên Fig. 4, vai trò của bộ lọc thứ ba 31 để lọc khoảng thời gian tiếp theo của khoảng thời gian hiện tại được thực hiện bởi phần lọc thứ hai 24. (Trong các ví dụ, trong đó 100% khoảng thời gian cập nhật được lọc hai lần bởi các phần tử 22 và 24, các bộ chuyển hướng luôn trong cùng trạng thái cho phép cả hai phần tử 22 và 24 thực hiện lọc.)

Fig. 5a thể hiện hệ thống 50. Hệ thống 50 có thể bao gồm bộ lọc thứ nhất 51 (mà có thể là, ví dụ, bộ lọc 12 được tạo cấu hình như trên Fig. 1 hoặc 1a), bộ lọc thứ hai 52 (ví dụ, có thể là bộ lọc 14 được tạo cấu hình như trên Fig. 1 hoặc 1a) và/hoặc bộ lọc thứ ba 53 và/hoặc bộ lọc thứ tư 54 (có thể là tùy chọn). Trong các ví dụ, một số bộ lọc này có thể giống nhau (ví dụ: thu được với cùng một phần cứng) và chỉ được phân biệt với nhau bằng đầu vào và/hoặc các tham số. Bộ chọn 55 có thể hướng phép biểu diễn tín hiệu 11a của tín hiệu thông tin đầu vào 11 đến bất kỳ bộ lọc 51-54 nào trên cơ sở các giá trị của phép biểu diễn tín hiệu 11a và/hoặc dữ liệu điều khiển 11b.

Trong một số ví dụ, các bộ lọc thứ nhất, thứ hai và thứ ba 51, 52 và 53 thu được như trên Fig. 4. Do đó, bộ lọc thứ hai 52 cũng có thể thực hiện các hoạt động của bộ lọc thứ ba 53.

Bộ lọc thứ nhất 51 có thể triển khai bộ lọc thứ nhất 12 và/hoặc phần lọc thứ nhất 22 và được sử dụng để lọc khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại với các tham số (ban đầu được liên kết với khoảng thời gian cập nhật thứ $(k-1)$ trước đó) mà chuyển từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn. Bộ lọc thứ hai 52 có thể triển khai bộ lọc thứ hai 14 và/hoặc phần lọc thứ hai 24 và được sử dụng để lọc khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại với các

tham số (được liên kết thực tế với khoảng thời gian cập nhật thứ k hiện tại) mà chuyển từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn. Bộ lọc thứ ba 53 có thể triển khai phần lọc thứ ba 31 cho khoảng thời gian tiếp theo của khoảng thời gian cập nhật hiện tại.

Bộ lọc thứ tư 54 có thể triển khai bộ lọc để lọc khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại với các tham số thu được bằng cách nội suy các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó và các tham số cho khoảng thời gian cập nhật thứ k hiện tại.

Bộ chọn 55 có thể hoạt động:

- trong khoảng thời gian phụ ban đầu (T_l), bằng cách lựa chọn trong số:
 - hoạt động lọc dựa trên hành động kết hợp của bộ lọc thứ nhất 51 và bộ lọc thứ hai 52;
 - hoạt động lọc dựa trên bộ lọc thứ ba 53; và
 - hoạt động lọc dựa trên bộ lọc thứ tư 54;
- trong khoảng thời gian tiếp theo, bằng cách sử dụng bộ lọc thứ ba 53.

Với việc tham chiếu đến quyết định cho khoảng thời gian phụ ban đầu, bộ chọn 55 có thể hoạt động, ví dụ, bằng cách sử dụng ngưỡng thứ nhất và/hoặc thứ hai và/hoặc các điều kiện (ví dụ: các điều kiện về độ khuếch đại của tín hiệu trong các khoảng thời gian cập nhật tiếp theo). Ví dụ: bộ chọn 55 có thể chọn:

- hành động kết hợp của bộ lọc thứ nhất 51 và bộ lọc thứ hai 52 khi khoảng cách giữa các tham số của khoảng thời gian cập nhật thứ k hiện tại và khoảng thời gian cập nhật thứ $(k-1)$ trước đó là cao, ví dụ: trên ngưỡng thứ hai;
- hành động của bộ lọc thứ tư 54 chỉ khi khoảng cách giữa các tham số của khoảng thời gian cập nhật thứ k hiện tại và các tham số của khoảng thời gian cập nhật thứ $(k-1)$ trước đó là nhỏ, ví dụ: dưới ngưỡng thứ hai; và/hoặc

○ hành động của bộ lọc thứ ba 53 chỉ khi khoảng cách giữa các tham số nhỏ hơn ngưỡng thứ nhất (có thể nhỏ hơn ngưỡng thứ hai) và/hoặc khi các tham số của khoảng cập nhật thứ k hiện tại giống với các tham số của khoảng thời gian cập nhật $(k-1)$ trước đó.

Ngưỡng thứ hai có thể được thiết lập hoạt động, ví dụ, như là cực tiểu giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó,

Theo cách bổ sung hoặc thay thế, cũng có thể sử dụng điều kiện dựa trên độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật trước đó, để sử dụng bộ lọc thứ tư 54 khi khoảng cách giữa độ khuếch đại tại khoảng thời gian cập nhật được xác định (hiện tại) và độ khuếch đại tại khoảng thời gian cập nhật trước đó nhỏ hơn ngưỡng thứ nhất và/hoặc thứ hai. Theo đó, ngưỡng thứ hai cũng như điều kiện về độ khuếch đại có thể được sửa đổi trong thời gian thực, ví dụ: để có được cách lọc tốt hơn.

Fig. 5b thể hiện cách triển khai cho bộ lọc sau LTP trong đó bộ lọc thứ tư 54 không được sử dụng (ví dụ, nó có thể bao gồm các đơn vị 12 và 14 được tạo kết cấu như trên Fig. 1 hoặc 1a).

Fig. 6 thể hiện phương pháp 60 trong đó, "UI" đề cập đến "khoảng thời gian cập nhật" và "SI" đề cập đến "khoảng thời gian phụ". Theo phương pháp 60, có thể thu được tín hiệu thông tin đầu vào 11 (x) (ví dụ: từ bộ mã hóa). Cụ thể, tín hiệu 11 có thể bao gồm phép biểu diễn tín hiệu 11a (ví dụ: được liên kết với tín hiệu âm thanh được lọc) và dữ liệu điều khiển 11b (ví dụ, có thể bao gồm các tham số bộ lọc liên kết với khoảng thời gian cập nhật hiện tại).

Tại bước S61, có thể thu được tín hiệu thông tin đầu vào (ví dụ: x, 11) cho khoảng thời gian cập nhật thứ k hiện tại (đã xác định) cùng với các tham số liên quan đến khoảng cập nhật thứ k hiện tại.

Sau đó, các hoạt động của bộ lọc thứ nhất 61 có thể được thực hiện (ví dụ: bởi bất kỳ thành phần nào 12, 22, 51) cho khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại, bằng cách tái chuyển giá trị (chỉ số) n trong số nhiều mẫu của khoảng thời gian phụ ban đầu. Tại S611, mẫu thứ nhất của khoảng thời gian phụ ban

đầu có thể được xem xét bằng cách khởi tạo biến n (như $n = 0$). Tại S612, giá trị trung gian $y'[n]$ thu được bằng cách sử dụng các tham số liên quan đến khoảng cập nhật thứ $(k-1)$ trước đó. Tại 613 (mẫu cuối cùng của SI ban đầu?), người ta kiểm tra xem n có đạt được giá trị liên quan đến mẫu cuối cùng của khoảng thời gian phụ ban đầu hay không (ví dụ: nó được kiểm tra xem n có phải là chỉ số cuối cùng của khoảng thời gian phụ ban đầu không). Nếu n đã đạt đến giá trị cuối cùng (chỉ số) của khoảng thời gian phụ ban đầu, các hoạt động của bộ lọc thứ nhất 61 được kết thúc và các hoạt động của bộ lọc thứ hai 62 được bắt đầu. Mặt khác, tại S614 (thay đổi các tham số dọc theo SI ban đầu của UI thứ k hiện tại từ trạng thái lọc cao hơn sang trạng thái lọc thấp hơn), các tham số được thay đổi để chuyển từ trạng thái lọc cao hơn sang trạng thái thấp hơn (ví dụ: bằng cách giảm các hệ số tại khối 23b của Fig. 2). Tại S615, mẫu mới được xem xét (bằng cách cập nhật chỉ số, ví dụ: bởi $n = n + 1$) và bước S612 được lặp lại cho chỉ số mới n .

Sau đó, các hoạt động của bộ lọc thứ hai 62 có thể được thực hiện (ví dụ: bởi bất kỳ thành phần nào 14, 24, 52) cho khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại, bằng cách tái chuyển giá trị (chỉ số) n trong số nhiều mẫu của khoảng thời gian phụ ban đầu. Tại S621, mẫu thứ nhất (của r $n=0$) của khoảng thời gian phụ ban đầu có thể được xem xét bằng cách khởi tạo biến n đến 0. Tại S612, giá trị đầu ra đã lọc $y[n]$ thu được bằng cách sử dụng các tham số liên quan đến khoảng cập nhật thứ k hiện tại. Tại 623 (mẫu cuối cùng của SI ban đầu?), người ta kiểm tra xem chỉ số n có đạt được giá trị liên quan đến mẫu cuối cùng của khoảng thời gian phụ ban đầu hay không (ví dụ: nó được kiểm tra xem n có phải là chỉ số cuối cùng của khoảng thời gian phụ ban đầu không). Nếu chỉ số n đã đạt đến giá trị cuối cùng của khoảng thời gian phụ ban đầu, các hoạt động của bộ lọc thứ hai 62 được kết thúc và các hoạt động của bộ lọc thứ ba 63 được bắt đầu. Mặt khác, tại S624 ("thay đổi các tham số dọc theo SI ban đầu của UI thứ k hiện tại từ trạng thái lọc thấp hơn sang trạng thái lọc cao hơn"), các tham số được thay đổi để chuyển từ trạng thái lọc thấp hơn sang cao hơn trạng thái lọc cao hơn (ví dụ: mở rộng các giá trị từ 0 hoặc giá trị gần 0 đến giá trị xa 0 hơn, ví dụ: bằng cách tăng giá trị của các hệ số tại khối 23a). Tại S625, mẫu mới được xem xét ($n = n + 1$) và bước S612 được gọi ra.

Sau đó, các hoạt động của bộ lọc thứ ba 63 được thực hiện (ví dụ: bởi bất kỳ thành phần nào 31, 24, 53) cho khoảng thời gian phụ tiếp theo (ví dụ, cuối cùng) của khoảng thời gian cập nhật thứ k hiện tại, bằng cách tái chuyển giá trị (chỉ số) n trong số nhiều mẫu của khoảng thời gian tiếp theo. Tại S632, giá trị đầu ra đã lọc $y[n]$ thu được bằng cách sử dụng các tham số liên quan đến khoảng cập nhật thứ k hiện tại. Tại 633, xác định xem chỉ số n có đạt được giá trị liên quan đến mẫu cuối cùng của khoảng thời gian cập nhật thứ k hiện tại hay không (ví dụ: kiểm tra xem n có phải là chỉ số cuối cùng của khoảng thời gian cập nhật hay không). Nếu n đã đạt đến giá trị cuối cùng của khoảng thời gian cập nhật, hoạt động của bộ lọc thứ ba 63 được kết thúc. Nếu không thì, tại S635, mẫu mới được xem xét (bằng cách cập nhật chỉ số, ví dụ: bởi $n = n + 1$) và bước S632 được lặp lại cho chỉ số mới n .

Tại cuối phương pháp, tất cả các giá trị $y[n]$ của tín hiệu đầu ra được lọc đã thu được. Giá trị của chỉ số k có thể được cập nhật tại S64. Bước S61 có thể được gọi lại trong khoảng thời gian cập nhật thứ $(k + 1)$.

Cần lưu ý rằng bước S61 không thực sự cần thiết sau bất kỳ bước S611-S63 nào. Trong một số ví dụ, tín hiệu thông tin đầu vào tại khoảng thời gian cập nhật thứ k hiện tại cũng có thể thu được trong quá trình xử lý các hoạt động 61-63. Trong các ví dụ, tín hiệu thông tin đầu vào tại khoảng thời gian cập nhật thứ k hiện tại có thể đã thu được một ưu tiên.

Trong các ví dụ trong đó các hoạt động lọc thứ nhất và thứ hai được thực hiện cho 100% khoảng thời gian cập nhật, các hoạt động lọc thứ ba 63 không được thực hiện.

Fig. 7 thể hiện phương pháp 70 ("UI" đề cập đến "khoảng thời gian cập nhật" và "SI" đề cập đến "khoảng thời gian phụ"). Phương pháp 70 có thể bao gồm bước S71 ("thu được tín hiệu thông tin đầu vào x tại UI thứ k và các tham số liên quan đến UI thứ k "), trong đó, trong khoảng thời gian cập nhật thứ k hiện tại, tín hiệu x và các tham số liên quan đến tín hiệu x tại khoảng thời gian cập nhật thứ k hiện tại thu được.

Tại bước S72 ("là các tham số cho UI thứ k và các tham số cho UI thứ $(k-1)$ giống nhau hoặc là khoảng cách của chúng trong ngưỡng đầu thứ nhất?), so sánh thứ nhất được thực hiện: các tham số cho khoảng thời gian cập nhật thứ k hiện tại là được so sánh với

các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó (ví dụ: bởi bộ chọn 55). Nếu các tham số giống nhau hoặc khoảng cách giữa tham số của khoảng thời gian cập nhật thứ k và tham số của khoảng cập nhật thứ $(k-1)$ nằm trong ngưỡng thứ nhất (thấp), tại S73 (thực hiện các hoạt động lọc thứ ba 63 mà không làm mờ các hệ số dọc theo toàn bộ UI thứ k) các hoạt động lọc thứ ba 63 được thực hiện bằng cách sử dụng, cho khoảng thời gian cập nhật thứ k hiện tại (cả khoảng thời gian phụ ban đầu và khoảng thời gian phụ tiếp theo), các tham số liên quan đến khoảng thời gian cập nhật thứ k hiện tại (vì các tham số này giống nhau hoặc gần như giống nhau cho hai khoảng thời gian cập nhật liên tiếp, không cần thiết phải làm mờ hoặc làm nhẵn chúng: do đó, có thể áp dụng bộ lọc thứ ba 31 hoặc 53 cho toàn bộ khoảng thời gian cập nhật, mà không phân biệt khoảng thời gian phụ ban đầu với khoảng thời gian phụ tiếp theo). Sau đó, tại S77, giá trị của k được cập nhật và khoảng thời gian cập nhật mới thứ $(k + 1)$ bây giờ có thể được lọc. Quyết định này có thể được đưa ra, ví dụ, bởi bất kỳ bộ chọn nào được thể hiện, ví dụ, trên Fig. 4 và 5.

Nếu khoảng cách giữa các tham số vượt quá ngưỡng thứ nhất, có thể thực hiện kiểm tra thứ hai: ví dụ: tại S74 ("có phải khoảng cách giữa các tham số cho UI thứ k và khoảng cách giữa các tham số cho UI thứ $(k-1)$ nằm trong ngưỡng thứ hai không? $g_k=0$ hoặc $g_{k-1}=0$?") các tham số cho khoảng thời gian cập nhật thứ k hiện tại được so sánh với các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó (ví dụ: tại bộ chọn 55). Các tham số được kiểm tra tại S74 cũng có thể khác với các tham số được kiểm tra tại S72. Nếu khoảng cách giữa các tham số cho khoảng thời gian cập nhật thứ k hiện tại và khoảng cách giữa các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó nằm trong ngưỡng thứ hai (mà có thể cao hơn ngưỡng thứ nhất), tại S75 (thực hiện các hoạt động lọc thứ tư nội suy các tham số"), hoạt động lọc thứ tư (ví dụ, bởi bộ lọc thứ tư 54) có thể được hoạt động tại khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật thứ k hiện tại. Trong trường hợp này, các tham số được áp dụng cho khoảng thời gian phụ ban đầu có thể thu được bằng cách nội suy (và/hoặc lấy trung bình) các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó và các tham số cho khoảng thời gian cập nhật thứ k hiện tại. Sau đó, bước lọc thứ ba 63 có thể được hoạt động tại S75' (bước lọc

thứ 3 trên SI tiếp theo) cho khoảng thời gian tiếp theo. Sau đó, tại S77, (“ $k=k+1$ ”), k được cập nhật và khoảng thời gian cập nhật mới thứ $(k + 1)$ bây giờ có thể được lọc.

Nếu tại S74, người ta xác minh rằng khoảng cách giữa các tham số cho khoảng thời gian cập nhật thứ k hiện tại và khoảng cách giữa các tham số cho khoảng thời gian cập nhật thứ $(k-1)$ trước đó vượt quá ngưỡng thứ hai, tại bước S76 (thực hiện các hoạt động lọc thứ nhất và thứ hai 61 và 62 trên SI ban đầu của UI thứ k và thực hiện các hoạt động lọc thứ ba 63 trên SI tiếp theo của UI thứ k), các hoạt động lọc thứ nhất, thứ hai và thứ ba 61, 62, 63 có thể được hoạt động (ví dụ: bởi các phân tử 12, 14, 22, 24, 31, 51, 52 và/hoặc 53): do đó, khoảng thời gian phụ thứ nhất của khoảng thời gian cập nhật hiện tại có thể được lọc đầu tiên bởi bất kỳ 12, 22, 51 và/hoặc 61 và sau đó bởi bất kỳ 14, 24, 52 và/hoặc 62, và khoảng thời gian phụ tiếp theo được lọc bằng cách sử dụng bất kỳ 31, 53 và/hoặc 63. Sau đó, tại S77, k được cập nhật và khoảng thời gian cập nhật mới thứ $(k + 1)$ có thể được lọc.

Trong các ví dụ, ít nhất điều kiện khác có thể được thiết lập, theo cách bổ sung hoặc thay thế (trong một số trường hợp, tại bước S74). Trong một số ví dụ được cung cấp rằng, để bắt đầu bước S76, điều kiện phải được xác minh. Trong các ví dụ, điều kiện tại S74 có thể bao gồm ít nhất một trong số:

- Nếu độ khuếch đại của khoảng thời gian cập nhật (hiện tại) được xác định bằng 0 ($g_k = 0$), thì bộ lọc thứ nhất được sử dụng, bộ lọc thứ hai không được sử dụng và bộ lọc thứ ba không được sử dụng (và, trong trường hợp được cung cấp, bộ lọc thứ tư không được sử dụng).

- Nếu độ khuếch đại của khoảng thời gian cập nhật trước đó bằng 0 ($g_{k-1}=0$), thì bộ lọc thứ nhất không được sử dụng, bộ lọc thứ hai được sử dụng, bộ lọc thứ ba được sử dụng (và, trong trường hợp được cung cấp, bộ lọc thứ tư không được sử dụng).

- Nếu cả độ khuếch đại của các khoảng thời gian cập nhật trước đó và hiện tại khác với 0 ($g_{k-1} \neq 0$ and $g_k \neq 0$) thì những gì được sử dụng phụ thuộc vào các tham số khác (ví dụ, trong một số ví dụ, có thể xem xét sự khác biệt của số nguyên và/hoặc các phần thập phân của độ trễ tần số cơ bản).

Trong một số ví dụ, bước S74 có thể được thực hiện trước bước S72.

Trong một số ví dụ, chỉ có một so sánh được thực hiện. Do đó, có những ví dụ, không có các bước S72 và S73 và có những ví dụ không có các bước S74 và S75.

Trong một số ví dụ, chỉ có một so sánh được thực hiện. Do đó, có những ví dụ, không có các bước S72 và S73 và có những ví dụ không có các bước S74 và S75.

Trong các ví dụ, (các) ngưỡng thứ nhất và/hoặc thứ hai là (hoặc các điều kiện khác về các tham số, ví dụ: về độ khuếch đại) cho các bước thứ nhất và/hoặc thứ hai S72, S74, có thể thu được trong thời gian thực, ví dụ, từ các giá trị của các tham số.

Trong một số ví dụ liên quan đến bộ lọc LTP, ngưỡng thứ hai có thể là ngưỡng khoảng cách độ trễ tần số cơ bản được xác định để sử dụng bộ lọc thứ tư 54 (và/hoặc nội suy) khi khoảng cách (ví dụ: chênh lệch modulo) giữa các tham số của khoảng thời gian cập nhật thứ k hiện tại và các tham số của khoảng thời gian cập nhật thứ (k-1) trước đó nhỏ hơn khoảng tối thiểu giữa phần nguyên của độ trễ tần số cơ bản liên quan đến khoảng thời gian cập nhật thứ k và phần nguyên của độ trễ tần số cơ bản liên quan đến khoảng thời gian cập nhật thứ (k-1). Do đó, so sánh thứ hai tại bước S74 có thể là:

$$|T_{int,k} - T_{int,k-1}| < \min(T_{int,k}, T_{int,k-1})$$

trong đó $T_{int,k}$ và $T_{int,k-1}$ là các phần nguyên của độ trễ tần số cơ bản lần lượt tại các khoảng thời gian cập nhật k và k-1. Theo đó, so sánh thứ hai tại S74, trong một số ví dụ, có thể kiểm tra xem cả hai điều kiện sau có áp dụng hay không:

$$T_{int,k} < 2T_{int,k-1}$$

$$T_{int,k} > \frac{1}{2}T_{int,k-1},$$

Do đó, so sánh thứ hai tại S74 sao cho, để thực hiện quá trình lọc bằng bộ lọc thứ tư 54, cần thiết rằng phần nguyên của tần số cơ bản không tăng (từ khoảng thời gian cập nhật thứ (k-1) đến khoảng thời gian cập nhật thứ k) trong hơn 100% cũng như không giảm hơn 50%: nghĩa là không có tần số cơ bản gấp đôi hoặc giảm một nửa giữa khoảng thời gian cập nhật trước đó và khoảng thời gian cập nhật hiện tại.

Tương tự, so sánh thứ hai tại S74 sao cho bước lọc thứ nhất và thứ hai (ví dụ: sử dụng bất kỳ yếu tố nào 12, 14, 51, 52, v.v.) có thể được kích hoạt khi được xác minh rằng

$$|T_{int,k} - T_{int,k-1}| \geq \min(T_{int,k}, T_{int,k-1})$$

đó là cách để nói rằng ít nhất một trong hai điều kiện sau được xác minh:

$$T_{int,k} \geq 2T_{int,k-1}$$

$$T_{int,k} \leq \frac{1}{2}T_{int,k-1},$$

Do đó, các bộ lọc thứ nhất và thứ hai có thể được kích hoạt (ví dụ: bởi bộ chọn 55) khi phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật thứ k hiện tại thay đổi rất nhiều so với phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật thứ (k - 1) trước đó.

Điều kiện khác có thể được thiết lập. Ví dụ: bước S74 có thể cung cấp rằng, để thực hiện các hoạt động lọc thứ nhất và thứ hai 61 và 62 (ví dụ: bằng bộ lọc thứ nhất và thứ hai 12 và 14), ít nhất một điều kiện sau phải được xác minh:

$$g_k = 0$$

$$g_{k-1} = 0$$

Khi $g_{k-1} = 0$, hiệu quả có thể thu được tương tự như bỏ qua bộ lọc thứ nhất. Khi $g_k = 0$, hiệu quả có thể thu được tương tự như bỏ qua bộ lọc thứ hai.

Một điều kiện có thể là, nếu cả hai điều kiện sau được xác minh:

$$g_k \neq 0$$

$$g_{k-1} \neq 0$$

trong trường hợp đó, sự khác biệt giữa các phần nguyên của độ trễ tần số cơ bản (của khung hiện tại và khung trước đó) được kiểm tra tại S74 (ví dụ, như đã thảo luận tại trên).

Trong ví dụ này có thể thấy rằng:

- 1) nếu các tham số giữa khoảng thời gian cập nhật hiện tại và khoảng thời gian trước đó là như nhau, thì trong khoảng thời gian cập nhật hiện tại, sử dụng cùng một bộ lọc của khoảng thời gian cập nhật trước đó (bộ lọc thứ ba 53);
- 2) nếu các tham số của khoảng thời gian cập nhật hiện tại và các tham số của khoảng thời gian cập nhật trước đó cực kỳ khác nhau hoặc nếu ít nhất một trong số các độ khuếch đại bằng 0, thì nên sử dụng bộ lọc thứ nhất và thứ hai (12, 24, 14, 24, 51, 52, 61, 62);
- 3) nếu độ khuếch đại của khoảng thời gian cập nhật hiện tại và trước đó đều khác 0, thì xác định được từ độ trễ tần số cơ bản mà bộ lọc nên sử dụng.

Đáng chú ý, (2) tăng chất lượng so với (1) và (3). (2) có độ phức tạp thấp hơn tình trạng kỹ thuật.

Trong một số ví dụ, bộ lọc thứ tư 54 không được sử dụng và do đó, xác minh thứ hai tại S74 không được thực hiện và chỉ có thể thực hiện so sánh với ngưỡng cực kỳ nhỏ (hoặc so sánh về giá trị chính xác).

Các ví dụ khác (ví dụ: bộ lọc không phải LTP) có thể dựa trên các tham số khác. Tuy nhiên, phương pháp hiện tại thực hiện cho bất kỳ bộ lọc IIR nào.

Nói chung, nếu có sự khác biệt về tham số, thì bộ lọc thứ nhất và thứ hai được sử dụng. Nếu không thì, bộ lọc thứ ba được sử dụng trong khoảng thời gian phụ ban đầu.

Giải pháp hiện tại có thể được sử dụng ví dụ khi các tham số LPC thay đổi trong bộ mã hóa - giải mã dự báo tuyến tính kích thích mã (Code-Excited Linear Prediction-CELP). Theo cách đó, sự không liên tục tồn tại ngay cả sau khi phép nội suy tần số phổ dòng (line spectral frequencies LSF) dựa trên khung con, trong CELP có thể được xử lý. Một ví dụ khác mà kỹ thuật này có thể được sử dụng là lọc để tăng cường định dạng trong các bộ mã hóa-giải mã CELP.

Các hình vẽ Fig. 8 và 9 thể hiện các phần tử của (các) hệ thống mã hóa/giải mã.

Fig. 8 thể hiện ví dụ về thiết bị 80 để mã hóa tín hiệu thông tin tại định dạng kỹ thuật số từ tín hiệu âm thanh 81.

Thiết bị 80 có thể bao gồm, ví dụ, bộ tiền lọc tùy chọn 81a có thể được hoạt động, ví dụ, như bất kỳ hệ thống 10, 30, 40 và 50 nào, và có thể thực hiện bất kỳ phương pháp nào nêu trên và/hoặc bên dưới. Trong các ví dụ khác, thiết bị 80 có thể được tránh.

Thiết bị 80 có thể bao gồm khối tạo cửa sổ 82. Thiết bị 80 có thể chứa bộ chuyển đổi miền 83 mà có thể chuyển đổi phép biểu diễn miền thời gian (TD) của tín hiệu thông tin thành biểu diễn miền tần số (FD) của tín hiệu thông tin. Ví dụ: bộ chuyển đổi 83 có thể là khối biến đổi cosin rời rạc cải biên (Modified Discrete Cosine Transform - MDCT) hoặc khối biến đổi sin rời rạc cải biên (Modified Discrete Sine Transform-MDST) 83 (hoặc khối được liên kết với phép biến đổi chồng khác), phía sau khối tạo cửa sổ 82, ví dụ, để chuyển đổi thành miền tần số (FD). Thiết bị 80 có thể bao gồm khối tạo hình nhiễu âm theo thời gian (TNS) 84 để kiểm soát hình dạng theo thời gian của nhiễu âm lượng tử hóa trong cửa sổ của phép biến đổi. Thiết bị 80 có thể bao gồm khối tạo hình nhiễu âm miền tần số (frequency domain noise shaper-FDNS) 85. Thiết bị 80 có thể chứa khối 87 để thu được các tham số FDNS phía sau khối tạo cửa sổ 82. Thiết bị 80 có thể bao gồm khối lượng tử hóa 86, cũng có thể bao gồm bộ mã hóa entropy. Thiết bị 80 có thể chứa khối TD/TD (bộ phát hiện tạm thời miền thời gian) 88.

Thiết bị 80 có thể chứa khối LTP 89 để thu được các tham số LTP (ví dụ: thông tin điều hòa, thông tin độ khuếch đại, thông tin tần số cơ bản như độ trễ tần số cơ bản, v.v.). Ít nhất một số tham số thu được từ khối LTP 89 có thể được sử dụng cho các thiết bị 10, 30, 40, 50 và/hoặc theo phương pháp 60 và/hoặc 70 cho mỗi khoảng thời gian cập nhật thứ k của tín hiệu. Ví dụ, các tham số cho khoảng thời gian cập nhật thứ k có thể là độ trễ tần số cơ bản và độ khuếch đại (mà một số trường hợp là tùy chọn và có thể được ước lượng tại phía bộ giải mã) liên quan đến tín hiệu tại khoảng thời gian cập nhật thứ k. Các hoạt động của khối LTP 89 có thể độc lập với các hoạt động của bộ tiền lọc 81a: bộ tiền lọc 81a cũng có thể không có mặt, nhưng khối LTP 89 có thể hoạt động chính xác bằng cách cung cấp các tham số cho phía bộ giải mã.

Tín hiệu có thể được mã hóa bởi bộ ghi dòng bit 89' và có thể được lưu trong bộ nhớ và/hoặc truyền đến bộ giải mã (ví dụ: theo cách không dây, ví dụ: sử dụng giao thức chuẩn như Bluetooth)

Fig. 9 thể hiện thiết bị 90 để mã hóa thông tin tín hiệu mà có thể thu được tín hiệu âm thanh kỹ thuật số (ví dụ: sử dụng đầu đọc dòng bit 91' được truyền từ hoặc được lưu trữ bởi bộ mã hóa, chẳng hạn như thiết bị 80). Thiết bị 90 có thể chứa ít nhất một trong các yếu tố 10, 12, 14, 20, 22, 24, 31, 40, 50 và/hoặc thực hiện bất kỳ phương pháp 60 và/hoặc 70 nào cho mỗi khoảng cập nhật thứ k của tín hiệu để cung cấp tín hiệu đầu ra được giải mã và lọc 15. Cụ thể, thiết bị 90 có thể chứa bộ lọc sau LTP 91 có thể thực hiện bất kỳ hành động lọc nào liên quan đến các yếu tố 12, 14, 22, 24, 31, 40, 51-54, 61-63 và 70. Thiết bị 90 có thể bao gồm khối lượng tử hóa 92, cũng có thể bao gồm bộ giải mã entropy. Thiết bị 90 có thể chứa khối FDNS 93, có thể nhận các tham số từ bộ giải mã tham số FDNS 94. Thiết bị 90 có thể chứa khối TNS 95 phía sau khối FDNS 93. Thiết bị 90 có thể chứa bộ chuyển đổi miền 96 mà có thể chuyển đổi phép biểu diễn miền thứ nhất (ví dụ: phép biểu diễn miền FD) của tín hiệu thông tin thành biểu diễn miền thứ hai (ví dụ: phép biểu diễn TD) của tín hiệu thông tin. Bộ chuyển đổi 96 có thể là khối MDCT nghịch đảo hoặc khối MDST nghịch đảo (hoặc khối được liên kết với phép biến đổi chùng khác) để chuyển đổi sang miền thời gian từ miền tần số. Thiết bị 90 có thể bao gồm khối tạo cửa sổ và xếp chồng và bổ sung (OLA) 97 có thể nhận các tham số từ khối TD/TD 88.

Ví dụ, bộ lọc sau LTP 91 có thể thu được phép biểu diễn kỹ thuật số 11a (x) của tín hiệu được lọc từ khối 97. Ví dụ, bộ lọc sau LTP 91 có thể thu được các hệ số 11b từ dòng bit.

Ít nhất một trong các hệ thống 80 và/hoặc 90 có thể thực hiện hoạt động phân tích (ví dụ: tại khối 89) để thu được các tham số liên quan đến khoảng thời gian cập nhật thứ k và/hoặc $(k + 1)$.

Tín hiệu thông tin đầu vào kỹ thuật số $x[n]$ (11a) có thể được lọc bằng bộ lọc thay đổi thời gian, mà các tham số của nó thay đổi tại khoảng thời gian cập nhật T (ví dụ, khoảng thời gian cập nhật thứ k hiện tại), tạo ra tín hiệu đầu ra được lọc $y[n]$. Khoảng thời gian cập nhật T cũng có thể là tín hiệu thích ứng và do đó T có thể thay đổi theo thời gian. Chúng ta có thể xem xét các bộ lọc có thể được biểu diễn dưới dạng bộ lọc đáp ứng xung vô hạn (IIR) bất biến thời gian tuyến tính (LTI) trong khoảng thời gian T . Khoảng thời gian T có thể là khung (ví dụ: bộ lọc hiện tại được thảo luận ở trên) hoặc

khung phụ của tín hiệu số. Chúng ta có thể sử dụng khung thời hạn và/hoặc khoảng thời gian cập nhật cho cả khung và khung phụ.

Bộ lọc LTI IIR (mà có thể cũng được biểu diễn như $H(z)$) có thể được thể hiện dưới dạng:

$$H(z) = \frac{1 + \sum_{i=0}^P b_i z^{-i}}{1 + \sum_{j=1}^Q a_j z^{-j}}$$

hoặc sử dụng phương trình chênh lệch tuyến tính

$$y[n] = x[n] + \sum_{i=0}^P b_i x[n-i] - \sum_{j=1}^Q a_j y[n-j]$$

Các hệ số b_i và a_j là các tham số bộ lọc (ví dụ, các tham số được lưu trữ trong các phần tử bộ nhớ 21a và 21b). Bộ lọc LTI IIR có thể được xác định duy nhất bởi các hệ số (tham số) của nó. Fig. 10 thể hiện khoảng thời gian cập nhật T liên kết với bộ lọc H_k (ví dụ: tại khoảng thời gian cập nhật thứ k hiện tại) và khoảng thời gian trước đó liên kết với bộ lọc H_{k-1} .

Chúng ta xem xét đến bộ lọc thay đổi theo thời gian mà là trong khoảng thời gian cập nhật k bằng với bộ lọc LTI IIR filter H_k :

$$H_k(z) = \frac{1 + \sum_{i=0}^P b_{k,i} z^{-i}}{1 + \sum_{j=1}^Q a_{k,j} z^{-j}}$$

$$y[n] = x[n] + \sum_{i=0}^P b_{k,i} x[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j], kT \leq n < (k+1)T$$

Thay vì ngay lập tức thay đổi các tham số bộ lọc tại biên giữa các khoảng thời gian cập nhật $k-1$ và k , chúng ta xử lý một phần tại phần bắt đầu khoảng thời gian cập nhật k bộ lọc thay đổi theo thời gian mới:

1. Nếu (bước S72) các tham số bộ lọc là như nhau (của sự khác biệt là cực kỳ nhỏ), việc lọc với bộ lọc H_k được thực hiện (bước S73);
2. Nếu (bước S74) khoảng cách giữa các tham số bộ lọc nhỏ (ví dụ: trong ngưỡng thứ hai được kiểm tra tại S74), thì các tham số bộ lọc được nội suy

theo mẫu (S75) và phần bắt đầu của khoảng thời gian cập nhật k được lọc bằng cách sử dụng tham số nội suy;

3. Nếu (bước S74) khoảng cách giữa các tham số bộ lọc là lớn (ví dụ: lớn hơn ngưỡng thứ hai), (bước S76), phần bắt đầu của chiều dài T_l trước tiên được lọc bằng bộ lọc H'_{k-1} (ví dụ: tại các phần tử như 12, 22, 51) và sau đó bằng H'_k (ví dụ: tại các phần tử như 14, 24, 52) được xác định bởi:

$$y'[n] = x[n] + s_{k-1}[n] \left(\sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y'[n-j] \right), kT \leq n < kT + T_l$$

$$s_{k-1}[n] = 1 - \frac{n - kT}{T_l}, kT \leq n < kT + T_l$$

$$y[n] = y'[n] + s_k[n] \left(\sum_{i=0}^P b_{k,i} y'[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j] \right), kT \leq n < kT + T_l$$

$$s_k[n] = \frac{n - kT}{T_l} = 1 - s_{k-1}[n], kT \leq n < kT + T_l$$

$y'[n]$ ví dụ, có thể là đầu ra trung gian của bộ lọc thứ nhất 12 của Fig. 1. $s_{k-1}[n]$ có thể là các hệ số định tỉ lệ của phần tử 23b (Fig. 2) để giảm các giá trị tham số $a_{k-1,i}$ và $b_{k-1,i}$ (được lưu trữ trong phần tử lưu trữ 21b). $s_k[n]$ có thể là các hệ số định tỉ lệ được lưu trữ trong phần tử 23a để giảm giá trị các tham số của phần tử 23a (Fig. 2) để giảm các giá trị tham số a_k và b_k (mà được lưu trữ trong phần tử lưu trữ 21b).

Ví dụ của $s_k[n]$ và $s_{k-1}[n]$ được cung cấp trên Fig. 11, trong đó T đề cập đến khoảng thời gian cập nhật thứ k hiện tại. Các hoạt động lọc thứ nhất và thứ hai 61 và 62 có thể được áp dụng cho ít nhất là khoảng thời gian phụ ban đầu T_l , trong khi hoạt động lọc thứ ba 63 được áp dụng cho T_s . Như có thể thấy, trong T_l $s_{k-1}[n]$ giảm, trong khi $s_k[n]$ tăng dần: điều này là do các tham số của khoảng thời gian cập nhật thứ $(k-1)^{th}$ trước đó được áp dụng cho tín hiệu đầu vào giảm dần, trong khi các tham số của khoảng thời gian cập nhật thứ k^{th} hiện tại T được tăng dần đến một giá trị tối đa mà, tại khoảng thời gian tiếp theo T_s , là hằng số.. Theo đó, có thể có được sự chuyển đổi suôn sẻ từ khoảng thời gian cập nhật thứ $(k-1)$ sang khoảng thời gian cập nhật thứ k.

Trên Fig. 11, cũng có thể thấy nơi lọc thứ ba 62 (ví dụ, được vận hành bởi bộ lọc thứ ba 53) có thể được thực hiện. Bước lọc thứ ba có thể được xác định bởi:

$$y[n] = x[n] + \sum_{i=0}^P b_{k,i} x[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j], kT + T_l \leq n < (k+1)T$$

Hệ thống như được mô tả trong tài liệu [3] có thể được sử dụng làm cơ sở, bao gồm bộ phát hiện chuyển tiếp miền thời gian (TD TD), tạo cửa sổ, MDCT, TNS, FDNS, OLA, lượng tử hóa, bộ mã hóa số học và LTP sau lọc (xem các khối 82-89 và 92 -97 ở trên). Việc sửa đổi có thể được thực hiện trong bộ lọc sau LTP và do đó các chi tiết của LTP sẽ được mô tả.

Nói chung, LTP có thể được coi là bộ lọc sau điều hòa để lọc phép biểu diễn tín hiệu thông tin. Nó có thể dựa trên hàm chuyển bao gồm tử số và mẫu số, trong đó tử số có thể chứa giá trị độ khuếch đại được biểu thị bằng thông tin độ khuếch đại và mẫu số có thể bao gồm phần nguyên của độ trễ tần số cơ bản được biểu thị bằng thông tin độ trễ tần số cơ bản và bộ lọc đa chạm tùy thuộc vào phần thập phân của độ trễ tần số cơ bản. Theo các ví dụ, trong đó hàm chuyển của bộ lọc sau bao gồm, trong tử số, bộ lọc FIR đa chạm tiếp theo cho phần thập phân bằng không của độ trễ tần số cơ bản. Theo các ví dụ, mẫu số bao gồm sản phẩm giữa bộ lọc đa chạm và giá trị độ khuếch đại. Theo các ví dụ, tử số còn bao gồm sản phẩm của giá trị vô hướng thứ nhất và giá trị vô hướng thứ hai, trong đó mẫu số bao gồm giá trị vô hướng thứ hai và không bao gồm giá trị vô hướng thứ nhất, trong đó giá trị vô hướng thứ nhất và thứ hai được xác định và có các giá trị lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng 1, và trong đó giá trị vô hướng thứ hai nhỏ hơn giá trị vô hướng thứ nhất.

Ví dụ về bộ lọc LTP được cung cấp dưới đây (xem thêm hình 13).

Tại phía bộ mã hóa (ví dụ: thiết bị 80), tính toán tham số LTP có thể được sử dụng như được mô tả trong tài liệu [14]:

1. Ước lượng bước

Một độ trễ tần số cơ bản (phần nguyên + phần thập phân) trên mỗi khung được ước lượng (kích thước khung, ví dụ, 20 mili giây). Một khung có thể, ví dụ, là một

khoảng thời gian cập nhật. Điều này được thực hiện trong hai bước để giảm độ phức tạp và cải thiện độ chính xác ước lượng.

a. Ước lượng thứ nhất của phần nguyên của độ trễ tần số cơ bản

Quy trình phân tích tần số cơ bản mà tạo ra đường viền tiến hóa tần số cơ bản nhằm được sử dụng (ví dụ, sự phân tích tần số cơ bản vòng mở được mô tả trong tài liệu [15], giấy 6.6). Sự phân tích này thường được thực hiện trên cơ sở khung con (kích thước khung con, ví dụ, 10 mili giây), và sản xuất một số ước lượng độ trễ tần số cơ bản mỗi khung con. Lưu ý rằng các ước lượng độ trễ tần số cơ bản này không có bất kỳ phân thập phân nào và thường được ước lượng trên tín hiệu được lấy mẫu giảm (tốc độ lấy mẫu, ví dụ 6400 Hz). Tín hiệu được sử dụng có thể là bất kỳ tín hiệu âm thanh nào, ví dụ, tín hiệu đầu vào hoặc tín hiệu âm thanh được gán trọng số LPC như được mô tả trong tài liệu [15], giấy 6.5.

b. Làm mịn phần nguyên của độ trễ tần số cơ bản

Phần nguyên cuối cùng của độ trễ tần số cơ bản được ước lượng trên tín hiệu âm thanh $x[n]$ chạy tại tốc độ lấy mẫu bộ mã hóa lõi, mà thường cao hơn tốc độ lấy mẫu của tín hiệu được lấy mẫu giảm được sử dụng trong a. (ví dụ, 12,8kHz, 16kHz, 32kHz, ...). Tín hiệu $x[n]$ có thể là tín hiệu âm thanh bất kỳ, ví dụ, tín hiệu âm thanh được gán trọng số LPC.

Phần nguyên của độ trễ tần số cơ bản sau đó là độ trễ d_m mà cực đại hóa hàm tự tương quan.

$$C(d) = \sum_{n=0}^N x[n]x[n-d]$$

với d quanh độ trễ tần số cơ bản T được ước lượng trong bước 1.a.

$$T - \delta_1 \leq d \leq T + \delta_2$$

c. Ước lượng phân thập phân của độ trễ tần số cơ bản

Phân thập phân có thể được tìm ra bằng cách nội suy hàm tự tương quan $C(d)$ được tính trong bước 1.b. và lựa chọn độ trễ tần số cơ bản theo

phân số mà cực đại hóa hàm tự tương quan được nội suy. Phép nội suy có thể được thực hiện sử dụng bộ lọc đáp ứng xung hữu hạn (Finite impulse response-FIR) như được mô tả trong ví dụ, tài liệu [15], giấy 6.6.7.

2. Ước lượng độ khuếch đại và sự lượng tử hóa

Độ khuếch đại có thể được ước lượng trên tín hiệu âm thanh đầu vào tại tốc độ lấy mẫu bộ mã hóa lỗi, nhưng có thể cũng là bất kỳ tín hiệu âm thanh như tín hiệu âm thanh được gán trọng số LPC. Tín hiệu này được ghi chú là $y[n]$ và có thể là giống hoặc khác với $x[n]$.

Phép dự báo $y_P[n]$ của $y[n]$ được tìm ra đầu tiên bằng cách lọc $y[n]$ bằng bộ lọc sau

$$P(z) = B(z, T_{fr})z^{-T_{int}}$$

với T_{int} là phần nguyên của độ trễ tần số cơ bản (được ước lượng trong 1.b.) và $B(z, T_{fr})$ là bộ lọc FIR thông thấp mà các hệ số của nó phụ thuộc vào phần thập phân của độ trễ tần số cơ bản T_{fr} (được ước lượng trong 1.c.).

Một ví dụ của $B(z)$ khi độ phân giải độ trễ tần số cơ bản là $\frac{1}{4}$:

$$\begin{array}{ll} T_{fr} = \frac{0}{4} & B(z) = 0.0000z^{-2} + 0.2325z^{-1} + 0.5349z^0 + 0.2325z^1 \\ T_{fr} = \frac{1}{4} & B(z) = 0.0152z^{-2} + 0.3400z^{-1} + 0.5094z^0 + 0.1353z^1 \\ T_{fr} = \frac{2}{4} & B(z) = 0.0609z^{-2} + 0.4391z^{-1} + 0.4391z^0 + 0.0609z^1 \\ T_{fr} = \frac{3}{4} & B(z) = 0.1353z^{-2} + 0.5094z^{-1} + 0.3400z^0 + 0.0152z^1 \end{array}$$

Độ khuếch đại g sau đó được tính toán như sau:

$$g = \frac{\sum_{n=0}^{N-1} y[n]y_P[n]}{\sum_{n=0}^{N-1} y_P[n]y_P[n]}$$

và được giới hạn giữa 0 và 1.

Cuối cùng, độ khuếch đại được lượng tử hóa, ví dụ, bằng 2 bit, sử dụng ví dụ, phép lượng tử hóa đồng đều.

Bộ lọc sau LTP từ tài liệu [14] có thể được sử dụng (ví dụ về hàm chuyển được cung cấp trên Fig. 13.):

$$H(z) = \frac{1 - \alpha\beta gB(z, 0)}{1 - \beta gB(z, T_{fr})z^{-T_{int}}}$$

các tham số của nó có thể được xác định từ các tham số được ước lượng tại phía bộ mã hóa và được giải mã từ dòng bit. g là độ khuếch đại được giải mã, T_{int} và T_{fr} phần nguyên và phần thập phân của độ trễ tần số cơ bản được giải mã, α và β là hai vô hướng mà gán trọng số độ khuếch đại, và $B(z, T_{fr})$ là bộ lọc FIR thông thấp mà các hệ số của nó phụ thuộc vào phần thập phân của độ trễ tần số cơ bản được giải mã. Thứ tự và các hệ số của $B(z, T_{fr})$ có thể cũng phụ thuộc vào tốc độ truyền dữ liệu và tốc độ lấy mẫu đầu ra. Đáp ứng tần số khác nhau có thể được thiết kế và tinh chỉnh cho mỗi tổ hợp của tốc độ truyền dữ liệu và tốc độ lấy mẫu đầu ra.

Sự khác biệt trong bộ lọc sau LTP đến [14] có thể là sự chuyển từ khung này sang khung tiếp theo. Bộ lọc sau LTP được sử dụng trong phần kết thúc của khung $k - 1$ là H_{k-1} :

$$H_{k-1}(z) = \frac{1 - \alpha\beta g_{k-1}B(z, 0)}{1 - \beta g_{k-1}B(z, T_{fr,k-1})z^{-T_{int,k-1}}}$$

$$y[n] = x[n] - \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y[n - T_{int,k-1} - j] \right)$$

và trong phần kết thúc của khung k là H_k :

$$H_k(z) = \frac{1 - \alpha\beta g_k B(z, 0)}{1 - \beta g_k B(z, T_{fr,k})z^{-T_{int,k}}}$$

$$y[n] = x[n] - \beta g_k \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j] \right)$$

b_{-1}^0	0,0000
b_0^0	0,2325

b_1^0	0,5349
b_2^0	0,2325

với $b_j(T_{fr})$ được xác định trong bảng sau:

T_{fr}	0/4	$\frac{1}{4}$	2/4	3/4
$b_{-1}(T_{fr})$	0,0000	0,0152	0,0609	0,1353
$b_0(T_{fr})$	0,2325	0,3400	0,4391	0,5094
$b_1(T_{fr})$	0,5349	0,5094	0,4391	0,3400
$b_2(T_{fr})$	0,2325	0,1353	0,0609	0,0152

Trong phần bắt đầu của khung k (khung thứ k^{th} hoặc khoảng thời gian cập nhật hiện tại) có thể có ba khả năng:

1. (bước S73): Nếu các tham số là giống nhau, cụ thể: $g_k = g_{k-1}$, $T_{int,k} = T_{int,k-1}$, $T_{fr,k} = T_{fr,k-1}$, phần bắt đầu của khung k được lọc với H_k ;
2. (bước S75): Nếu sự khác biệt giữa các tham số là nhỏ, ví dụ, $|T_{int,k} - T_{int,k-1}| < \min(T_{int,k}, T_{int,k-1})$ và $|g_k - g_{k-1}| < \max(g_k, g_{k-1})$, phần bắt đầu của chiều dài L của khung k được lọc bằng bộ lọc thay đổi theo thời gian sử dụng các tham số được nội suy:

$$y[n] = x[n] - \beta g'_k[n] \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T'_{fr,k}[n]) y[n - T'_{int,k}[n] - j] \right)$$

$$g'_k[n] = \left(1 - \frac{n}{L}\right) g_{k-1} + \frac{n}{L} g_k$$

$$T_{k-1} = T_{int,k-1} + T_{fr,k-1}/4$$

$$T_k = T_{int,k} + T_{fr,k}/4$$

$$T'_k[n] = \left(1 - \frac{n}{L}\right) T_{k-1} + \frac{n}{L} T_k$$

$$T'_{int,k}[n] = \lfloor T'_k[n] \rfloor$$

$$T'_{fr,k}[n] = T'_k[n] - T'_{int,k}[n]$$

$$0 \leq n < L$$

3. (bước S76): Nếu sự khác biệt giữa các tham số là lớn, phần bắt đầu của chiều dài L của khung k được lọc đầu tiên bằng bộ lọc H'_{k-1} :

$$y'[n] = x[n] - \left(1 - \frac{n}{L}\right) \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j] \right)$$

$$0 \leq n < L$$

và theo sau bởi H'_k :

$$y[n] = x[n] - \frac{n}{L} \beta g_k \left(\alpha \sum_{i=-1}^2 b_i^0 y'[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j] \right)$$

$$0 \leq n < L$$

Với các trường hợp trong đó độ phức tạp là quan trọng hơn chất lượng, khả năng 3 được sử dụng bất cứ khi nào ít nhất một trong số $g_k \neq g_{k-1}$, $T_{int,k} \neq T_{int,k-1}$, $T_{fr,k} \neq T_{fr,k-1}$ thỏa mãn. Một số trong những điều kiện này có thể quan trọng hơn một số điều kiện khác, theo các ví dụ cụ thể. Theo một số ví dụ, sự khác biệt trong tần số cơ bản là điều kiện quan trọng nhất được xác nhận, với lựa chọn giữa khả năng 2 và 3.

Một ví dụ về các ứng dụng của các phương trình trên cho bộ lọc sau LTP được cung cấp dưới đây với việc tham chiếu đến Fig. 7a và phương pháp 700. Các tham số có thể bao gồm độ khuếch đại g_k , phần nguyên của tần số cơ bản $T_{int,k}$ và phần thập phân của tần số cơ bản $T_{fr,k}$. Trong một số ví dụ này, bộ chọn 55 có thể hoạt động để:

- Nếu g_{k-1} (độ khuếch đại) bằng 0 và g_k cũng bằng 0 (S702), thì không có bước lọc (S704), bởi vì:

$$y'[n] = x[n] - \left(1 - \frac{n}{L}\right) \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j] \right) = x[n],$$

nhờ $g_{k-1} = 0$; và

$$y[n] = x[n] - \frac{n}{L} \beta g_k \left(\alpha \sum_{i=-1}^2 b_i^0 y'[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j] \right) = x[n],$$

nhờ $g_k = 0$;

(với việc tham chiếu đến Fig. 1a, cả hai bộ lọc thứ nhất và thứ hai 12 và 14 có thể được bỏ qua)

- Nếu $g_{k-1}=0$ và $g_k \neq 0$ (S706), sau đó (S708)

○ không có bước lọc thứ nhất, bởi vì $y'[n] = x[n] - \left(1 - \frac{n}{L}\right) \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j] \right) = x[n]$

nhờ $g_{k-1} = 0$;

○ có bước lọc thứ hai T_l ở dạng $y[n] = x[n] - \frac{n}{L} \beta g_k \left(\alpha \sum_{i=-1}^2 b_i^0 y'[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j] \right)$

○ có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo T_s (trong đó $T_l \neq T$) ở dạng

$$; y[n] = x[n] - \beta g_k \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j] \right)$$

(với tham chiếu đến Fig. 1a, chỉ có bộ lọc thứ nhất 12 được bỏ qua);

- Nếu $g_{k-1} \neq 0$ và $g_k = 0$ (S710), sau đó (S712)

○ có bước lọc thứ nhất T_l ở dạng $y'[n] = x[n] - \left(1 - \frac{n}{L}\right) \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j] \right)$

○ không có bước lọc thứ hai, bởi vì $y'[n] = x[n] - \left(1 - \frac{n}{L}\right) \beta g_{k-1} \left(\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j] \right) =$

$x[n]$

nhờ $g_k = 0$;

- không có bước lọc thứ ba, bởi vì

$$y[n] = x[n] - \beta g_k (\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j]) = x[n],$$

nhờ $g_k = 0$;

- (với tham chiếu đến Fig. 1a, chỉ có bộ lọc thứ hai 14 được bỏ qua);

- Nếu $g_{k-1} \neq 0$ và $g_k \neq 0$ (714), sau đó sự khác biệt của phần nguyên và phần thập phân của độ trễ tần số cơ bản được kiểm tra (S716, “ $T_{int,k} = T_{int,k-1}$ and $T_{fr,k-1} = T_{fr,k}$?”):

- Nếu phần nguyên và phần thập phân của độ trễ tần số cơ bản trong $k-1$ và k là như nhau ($T_{int,k} = T_{int,k}$ và $T_{fr,k-1} = T_{fr,k}$) sau đó (S718):

- không có bước lọc thứ nhất cũng như bước lọc thứ hai nhờ vào lựa chọn được vận hành bởi bộ chọn (ví dụ: tại bước S72);

- có bước lọc thứ ba dọc theo 100% khoảng thời gian cập nhật T , ở dạng $y[n] = x[n] - \beta g_k (\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j])$;

- ngoài ra nếu có sự khác biệt về phần nguyên hoặc phần thập phân của độ trễ tần số cơ bản (S720):

- có bước lọc thứ nhất T_l ở dạng $y'[n] = x[n] - (1 - \frac{n}{L}) \beta g_{k-1} (\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k-1}) y'[n - T_{int,k-1} - j])$

- có bước lọc thứ hai T_l ở dạng $y[n] = x[n] - \frac{n}{L} \beta g_k (\alpha \sum_{i=-1}^2 b_i^0 y'[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j])$

(với việc tham chiếu đến Fig. 1a, không có bộ lọc 12 và 14 được bỏ qua);

- có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo T_s (trong đó $T_l \neq T_s$) ở dạng $y[n] = x[n] - \beta g_k (\alpha \sum_{i=-1}^2 b_i^0 x[n-i] - \sum_{j=-1}^2 b_j(T_{fr,k}) y[n - T_{int,k} - j])$

Đáng chú ý, khi được xác định (ví dụ, bởi bộ chọn 55) rằng bước lọc thứ nhất và/hoặc thứ hai sẽ không được thực hiện (hoặc sẽ cung cấp, như đầu ra, cùng một giá trị của đầu vào, về cơ bản hoạt động như "bộ lọc nhận dạng", mà nói chung là vô dụng), có thể bỏ qua bộ lọc và/hoặc phân lọc vô dụng (ví dụ, như trên Fig. 1a). Theo đó, số lượng phép tính toán được giảm.

Thảo luận tại đây được cung cấp về hoạt động của các bộ lọc cho bộ lọc sau LTP. Tín hiệu được giải mã sau khi tổng hợp MDCT hoặc MDST (hoặc bất kỳ biến đổi chồng nào khác) có thể được lọc sau trong miền thời gian bằng cách sử dụng bộ lọc IIR mà tham số của nó phụ thuộc vào dữ liệu dòng bit LTPF, ví dụ, có thể là "pitch_index", và/hoặc "ltpf_active" (kích hoạt/hủy kích hoạt hoạt động lọc sau LTP). Để tránh gián đoạn khi các tham số thay đổi từ khung này sang khung tiếp theo, cơ chế chuyển đổi có thể được áp dụng vào phần tư thứ nhất của khung hiện tại.

Bộ lọc sau LTPF IIR có thể được thực hiện bằng cách sử dụng (xem thêm tại trên):

$$\widehat{x_{ltpf}}(n) = \hat{x}(n) - \sum_{k=0}^{L_{num}} c_{num}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}}\left(n - p_{int} + \frac{L_{den}}{2} - k\right)$$

với $\hat{x}(n)$ là tín hiệu đầu vào bộ lọc (nghĩa là tín hiệu được giải mã sau khi tổng hợp MDCT hoặc MDST) và $\widehat{x_{ltpf}}(n)$ là tín hiệu đầu ra bộ lọc.

Phần nguyên p_{int} và phần thập phân p_{fr} của độ trễ tần số cơ bản LTPF có thể được tính như sau. Đầu tiên, độ trễ tần số cơ bản (ví dụ: tại mức 12,8kHz) có thể được phục hồi bằng cách sử dụng

$$\text{pitch_int} = \begin{cases} \text{pitch_index} - 283 & \text{if } \text{pitch_index} \geq 440 \\ \left\lfloor \frac{\text{pitch_index}}{2} \right\rfloor - 63 & \text{if } 440 > \text{pitch_index} \geq 380 \\ \left\lfloor \frac{\text{pitch_index}}{4} \right\rfloor + 32 & \text{if } 380 > \text{pitch_index} \end{cases}$$

$$\text{pitch_fr} = \begin{cases} 0 & \text{if } \text{pitch_index} \geq 440 \\ 2 * \text{pitch_index} - 4 * \text{pitch_int} + 508 & \text{if } 440 > \text{pitch_index} \geq 380 \\ \text{pitch_index} - 4 * \text{pitch_int} + 128 & \text{if } 380 > \text{pitch_index} \end{cases}$$

$$\text{pitch} = \text{pitch_int} + \frac{\text{pitch_fr}}{4}$$

Độ trễ tần số cơ bản sau đó có thể được định tỉ lệ theo tỉ lệ lấy mẫu đầu ra f_s và được chuyển đổi thành các phần nguyên và phần thập phân bằng cách sử dụng

$$\begin{aligned} \text{pitch}_{f_s} &= \text{pitch} * \frac{f_s}{12800} \\ p_{up} &= \text{nint}(\text{pitch}_{f_s} * 4) \\ p_{int} &= \left\lfloor \frac{p_{up}}{4} \right\rfloor \\ p_{fr} &= p_{up} - 4 * p_{int} \end{aligned} \quad (1)$$

trong đó F_s là tốc độ lấy mẫu

Các hệ số bộ lọc $c_{num}(k)$ và $c_{den}(k, p_{fr})$ có thể được tính toán như sau

$$\begin{aligned} c_{num}(k) &= 0.85 * \text{gain}_{ltpf} * \text{tab_ltpf_num_fs}[k] \quad \text{for } k = 0..L_{num} \\ c_{den}(k, p_{fr}) &= \text{gain}_{ltpf} * \text{tab_ltpf_den_fs}[p_{fr}][k] \quad \text{for } k = 0..L_{den} \end{aligned}$$

với

$$\begin{aligned} L_{den} &= \max\left(4, \frac{f_s}{4000}\right) \\ L_{num} &= L_{den} - 2 \end{aligned}$$

và gain_{ltpf} và gain_{ind} có thể thu được, trong một số ví dụ, theo các quy trình như:

```
fs_idx = min(4, (f_s/8000-1));
if (nbits < 320 + fs_idx*80)
{
    gain_ltpf = 0.4;
    gain_ind = 0;
}
else if (nbits < 400 + fs_idx*80)
{
    gain_ltpf = 0.35;
    gain_ind = 1;
}
else if (nbits < 480 + fs_idx*80)
```

```

{
    gain_ltpf = 0.3;
    gain_ind = 2;
}
else if (nbits < 560 + fs_idx*80)
{
    gain_ltpf = 0.25;
    gain_ind = 3;
}
else
{
    gain_ltpf = 0;
}

```

các bảng `tab_ltpf_num_fs[k]` và `tab_ltpf_den_fs[pfr][k]` được xác định trước. Một số ví dụ có thể là (thay vì "fs", một bảng thông thực tế được sử dụng):

```

double tab_ltpf_num_8000[4][3] = {
    {6.023618207009578e-01,4.197609261363617e-01,-1.883424527883687e-02},
    {5.994768582584314e-01,4.197609261363620e-01,-1.594928283631041e-02},
    {5.967764663733787e-01,4.197609261363617e-01,-1.324889095125780e-02},
    {5.942410120098895e-01,4.197609261363618e-01,-1.071343658776831e-02}};
double tab_ltpf_num_16000[4][3] = {
    {6.023618207009578e-01,4.197609261363617e-01,-1.883424527883687e-02},
    {5.994768582584314e-01,4.197609261363620e-01,-1.594928283631041e-02},
    {5.967764663733787e-01,4.197609261363617e-01,-1.324889095125780e-02},
    {5.942410120098895e-01,4.197609261363618e-01,-1.071343658776831e-02}};
double tab_ltpf_num_24000[4][5] = {
    {3.989695588963494e-01,5.142508607708275e-01,1.004382966157454e-01,-
1.278893956818042e-02,-1.572280075461383e-03},
    {3.948634911286333e-01,5.123819208048688e-01,1.043194926386267e-01,-
1.091999960222166e-02,-1.347408330627317e-03},

```

```

    {3.909844475885914e-01,5.106053522688359e-01,1.079832524685944e-01,-
9.143431066188848e-03,-1.132124620551895e-03},
    {3.873093888199928e-01,5.089122083363975e-01,1.114517380217371e-01,-
7.450287133750717e-03,-9.255514050963111e-04}};
    double tab_ltpf_num_32000[4][7] = {
        {2.982379446702096e-01,4.652809203721290e-01,2.105997428614279e-
01,3.766780380806063e-02,-1.015696155796564e-02,-2.535880996101096e-03,-
3.182946168719958e-04},
        {2.943834154510240e-01,4.619294002718798e-01,2.129465770091844e-
01,4.066175002688857e-02,-8.693272297010050e-03,-2.178307114679820e-03,-
2.742888063983188e-04},
        {2.907439213122688e-01,4.587461910960279e-01,2.151456974108970e-
01,4.350104772529774e-02,-7.295495347716925e-03,-1.834395637237086e-03,-
2.316920186482416e-04},
        {2.872975852589158e-01,4.557148886861379e-01,2.172126950911401e-
01,4.620088878229615e-02,-5.957463802125952e-03,-1.502934284345198e-03,-
1.903851911308866e-04}};
    double tab_ltpf_num_48000[4][11] = {
        {1.981363739883217e-01,3.524494903964904e-01,2.513695269649414e-
01,1.424146237314458e-01,5.704731023952599e-02,9.293366241586384e-03,-
7.226025368953745e-03,-3.172679890356356e-03,-1.121835963567014e-03,-
2.902957238400140e-04,-4.270815593769240e-05},
        {1.950709426598375e-01,3.484660408341632e-01,2.509988459466574e-
01,1.441167412482088e-01,5.928947317677285e-02,1.108923827452231e-02,-
6.192908108653504e-03,-2.726705509251737e-03,-9.667125826217151e-04,-
2.508100923165204e-04,-3.699938766131869e-05},
        {1.921810055196015e-01,3.446945561091513e-01,2.506220094626024e-
01,1.457102447664837e-01,6.141132133664525e-02,1.279941396562798e-02,-
5.203721087886321e-03,-2.297324511109085e-03,-8.165608133217555e-04,-
2.123855748277408e-04,-3.141271330981649e-05},

```

```

{1.894485314175868e-01,3.411139251108252e-01,2.502406876894361e-
01,1.472065631098081e-01,6.342477229539051e-02,1.443203434150312e-02,-
4.254449144657098e-03,-1.883081472613493e-03,-6.709619060722140e-04,-
1.749363341966872e-04,-2.593864735284285e-05}};

double tab_ltpf_den_16000[4][5] = {
    {0.000000000000000e+00, 2.098804630681809e-01, 5.835275754221211e-01,
2.098804630681809e-01, 0.000000000000000e+00},
    {0.000000000000000e+00, 1.069991860896389e-01, 5.500750019177116e-01,
3.356906254147840e-01, 6.698858366939680e-03},
    {0.000000000000000e+00, 3.967114782344967e-02, 4.592209296082350e-01,
4.592209296082350e-01, 3.967114782344967e-02},
    {0.000000000000000e+00, 6.698858366939680e-03, 3.356906254147840e-01,
5.500750019177116e-01, 1.069991860896389e-01}};

double tab_ltpf_den_24000[4][7] = {
    {0.000000000000000e+00, 6.322231627323796e-02, 2.507309606013235e-01,
3.713909428901578e-01, 2.507309606013235e-01, 6.322231627323796e-02,
0.000000000000000e+00},
    {0.000000000000000e+00, 3.459272174099855e-02, 1.986515602645028e-01,
3.626411726581452e-01, 2.986750548992179e-01, 1.013092873505928e-01,
4.263543712369752e-03},
    {0.000000000000000e+00, 1.535746784963907e-02, 1.474344878058222e-01,
3.374259553990717e-01, 3.374259553990717e-01, 1.474344878058222e-01,
1.535746784963907e-02},
    {0.000000000000000e+00, 4.263543712369752e-03, 1.013092873505928e-01,
2.986750548992179e-01, 3.626411726581452e-01, 1.986515602645028e-01,
3.459272174099855e-02}};

double tab_ltpf_den_32000[4][9] = {
    {0.000000000000000e+00, 2.900401878228730e-02, 1.129857420560927e-01,
2.212024028097570e-01, 2.723909472446145e-01, 2.212024028097570e-01,
1.129857420560927e-01, 2.900401878228730e-02, 0.000000000000000e+00},

```

```

{0.0000000000000000e+00, 1.703153418385261e-02, 8.722503785537784e-02,
1.961407762232199e-01, 2.689237982237257e-01, 2.424999102756389e-01,
1.405773364650031e-01, 4.474877169485788e-02, 3.127030243100724e-03},
{0.0000000000000000e+00, 8.563673748488349e-03, 6.426222944493845e-02,
1.687676705918012e-01, 2.587445937795505e-01, 2.587445937795505e-01,
1.687676705918012e-01, 6.426222944493845e-02, 8.563673748488349e-03},
{0.0000000000000000e+00, 3.127030243100724e-03, 4.474877169485788e-02,
1.405773364650031e-01, 2.424999102756389e-01, 2.689237982237257e-01,
1.961407762232199e-01, 8.722503785537784e-02, 1.703153418385261e-02}};

double tab_ltpf_den_48000[4][13] = {
{0.0000000000000000e+00, 1.082359386659387e-02, 3.608969221303979e-02,
7.676401468099964e-02, 1.241530577501703e-01, 1.627596438300696e-01,
1.776771417779109e-01, 1.627596438300696e-01, 1.241530577501703e-01,
7.676401468099964e-02, 3.608969221303979e-02, 1.082359386659387e-02,
0.0000000000000000e+00},
{0.0000000000000000e+00, 7.041404930459358e-03, 2.819702319820420e-02,
6.547044935127551e-02, 1.124647986743299e-01, 1.548418956489015e-01,
1.767122381341857e-01, 1.691507213057663e-01, 1.352901577989766e-01,
8.851425011427483e-02, 4.499353848562444e-02, 1.557613714732002e-02,
2.039721956502016e-03},
{0.0000000000000000e+00, 4.146998467444788e-03, 2.135757310741917e-02,
5.482735584552816e-02, 1.004971444643720e-01, 1.456060342830002e-01,
1.738439838565869e-01, 1.738439838565869e-01, 1.456060342830002e-01,
1.004971444643720e-01, 5.482735584552816e-02, 2.135757310741917e-02,
4.146998467444788e-03},
{0.0000000000000000e+00, 2.039721956502016e-03, 1.557613714732002e-02,
4.499353848562444e-02, 8.851425011427483e-02, 1.352901577989766e-01,
1.691507213057663e-01, 1.767122381341857e-01, 1.548418956489015e-01,
1.124647986743299e-01, 6.547044935127551e-02, 2.819702319820420e-02,
7.041404930459358e-03}};

```

Năm trường hợp khác nhau có thể được xem xét:

1. Trường hợp thứ nhất: $ltpf_active = 0$ và $mem_ltpf_active = 0$ (“ mem_ltpf_active ” đề cập đến trạng thái kích hoạt/hủy kích hoạt trong khung trước đó):

$$\widehat{x_{ltpf}}(n) = \hat{x}(n) \quad for \quad n = 0.. \frac{N_F}{4}$$

trong đó N_F đề cập đến số lượng mẫu được xử lý trong một khung, kích thước khung a.k.a.

2. Trường hợp thứ hai: $ltpf_active = 1$ và $mem_ltpf_active = 0$

$$\widehat{x_{ltpf}}(n) = \hat{x}(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}} \left(n - p_{int} + \frac{L_{den}}{2} - k \right) \right] \\ for \quad n = 0.. \frac{N_F}{4}$$

3. Trường hợp thứ ba: $ltpf_active = 0$ và $mem_ltpf_active = 1$

$$\widehat{x_{ltpf}}(n) = \hat{x}(n) - \left(1 - \frac{n}{\frac{N_F}{4}} \right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x_{ltpf}} \left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k \right) \right] \\ for \quad n = 0.. \frac{N_F}{4}$$

với c_{num}^{mem} , c_{den}^{mem} , p_{int}^{mem} và p_{fr}^{mem} là các tham số bộ lọc được tính trong khung trước đó

4. Trường hợp thứ tư: $ltpf_active = 1$ và $mem_ltpf_active = 1$ và $p_{int} = p_{int}^{mem}$ và

$$p_{fr} = p_{fr}^{mem}$$

$$\widehat{x_{ltpf}}(n) = \hat{x}(n) - \sum_{k=0}^{L_{num}} c_{num}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}} \left(n - p_{int} + \frac{L_{den}}{2} - k \right) \\ for \quad n = 0.. \frac{N_F}{4}$$

5. Trường hợp thứ năm: $ltpf_active = 1$ và $mem_ltpf_active = 1$ và ($p_{int} \neq p_{int}^{mem}$ hoặc $p_{fr} \neq p_{fr}^{mem}$)

$$\widehat{x_{ltpf}}'(n) = \hat{x}(n) - \left(1 - \frac{n}{\frac{N_F}{4}}\right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x_{ltpf}}' \left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k \right) \right]$$

$for\ n = 0.. \frac{N_F}{4}$

$$\widehat{x_{ltpf}}(n) = \widehat{x_{ltpf}}'(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x_{ltpf}}'(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}} \left(n - p_{int} + \frac{L_{den}}{2} - k \right) \right]$$

$for\ n = 0.. \frac{N_F}{4}$

Ưu điểm của sáng chế được thảo luận tại đây. Các ví dụ trên ít phức tạp hơn so với các cách triển khai được sử dụng trước đây. Ví dụ, trong trường hợp ví dụ của bộ lọc sau LTP, ưu điểm về sự phức tạp so với phương pháp LPC được sử dụng trong tài liệu [13] là rõ ràng. So sánh với các phương pháp nhiễu xuyên kênh được sử dụng trong tài liệu [9], [10], [11] và [12] có ít hơn một thao tác cho mỗi mẫu. Để thấy điều này, lưu ý rằng $(1 - \frac{n}{L})\beta g_{k-1}$ và $\frac{n}{L}\beta g_k$ có thể được nhận ra bằng cách trừ hằng số $\frac{1}{L}\beta g_{k-1}$ khỏi βg_{k-1} và thêm hằng số mẫu $\frac{1}{L}\beta g_k$ vào βg_k theo mẫu. Cùng với lọc sử dụng H'_{k-1} và H'_k đây là số lượng hoạt động tương đương với bộ lọc bằng H_{k-1} và H_k sau đó là gán trọng số các tín hiệu được lọc như được sử dụng trong phương pháp nhiễu xuyên kênh. Phương pháp nhiễu xuyên kênh sau đó được tiếp tục với việc bổ sung các tín hiệu được gán trọng số, trong khi phương pháp được đề xuất đã tạo ra đầu ra.

Khi xác định được rằng bước lọc thứ nhất và/hoặc thứ hai sẽ không được thực hiện (hoặc sẽ cung cấp, như đầu ra, cùng một giá trị của đầu vào, về cơ bản hoạt động như "bộ lọc nhận dạng", mà nói chung là vô dụng), có thể bỏ qua bộ lọc và/hoặc phân lọc vô dụng (ví dụ, như trên Fig. 1a). Theo đó, số lượng phép tính toán được giảm.

Nói chung, tránh sự gián đoạn khi bộ lọc thay đổi tại biên khung từ H_{k-1} :

$$y[n] = x[n] + \sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y[n-j]$$

đến H_k :

$$y[n] = x[n] + \sum_{i=0}^P b_{k,i} x[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j],$$

bằng cách lọc phần đầu với chiều dài L của khung k bằng bộ lọc thay đổi theo thời gian H'_{k-1} :

$$y'[n] = x[n] + s_{k-1}[n] \left(\sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y'[n-j] \right)$$

$$s_{k-1}[n] = 1 - \frac{n}{L}, 0 \leq n < L$$

sau đó là bước lọc đầu ra trung gian $y'[n]$ của bộ lọc thay đổi theo thời gian H'_{k-1} với bộ lọc thay đổi theo thời gian H'_k :

$$y[n] = y'[n] + s_k[n] \left(\sum_{i=0}^P b_{k,i} y'[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j] \right)$$

$$s_k[n] = \frac{n}{L} = 1 - s_{k-1}[n], 0 \leq n < L$$

H'_{k-1} là bộ lọc thay đổi theo thời gian thay đổi từ bước lọc hoàn toàn với H_{k-1} đến không lọc. H'_k là bộ lọc thay đổi theo thời gian mà thay đổi từ không lọc đến lọc hoàn toàn H_k .

Ví dụ thứ nhất (dựa trên ví dụ của Fig. 5b nêu trên) tại đây được cung cấp trong mã giả:

Nếu $g_{k-1} == g_k$ và $g_k == 0$ thì không lọc.

Hoặc nếu $g_{k-1} = 0$ và $g_k \neq 0$, thì

{

- không có bước lọc thứ nhất
- có bước lọc thứ hai trong T_1
- có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo (trong đó

$T_1 \neq T$)

}

Hoặc nếu $g_{k-1} \neq 0$ và $g_k == 0$ thì

{

- có bước lọc thứ nhất trong T_1
 - không có bước lọc thứ hai
 - không có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo
- }

Hoặc nếu $g_{k-1} \neq 0$ và $g_k \neq 0$ thì chênh lệch phần nguyên và phần thập phân của độ trễ tần số cơ bản được kiểm tra

{

Nếu phần nguyên và phần thập phân của độ trễ tần số cơ bản trong $k-1$ và k là giống nhau ($\text{pitch_int}(k-1) == \text{pitch_int}(k) \ \&\& \ \text{pitch_fr}(k-1) == \text{pitch_fr}(k)$) thì

{

- không có bước lọc thứ nhất cũng như lọc thứ hai;
- có bước lọc thứ ba trong T_1 và trong T_s (tức là, dọc theo toàn bộ T)

}

Hoặc nếu có sự khác biệt trong phần nguyên hoặc trong phần thập phân của độ trễ tần số cơ bản:

{

- có bước lọc thứ nhất trong T_1
- có bước lọc thứ hai trong T_1
- có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo (trong đó $T_1 \neq T$)

}

}

Ví dụ thứ hai được cung cấp trong giả mã:

2.

Nếu $g_{k-1} = g_k$ và $g_k = 0$ thì không lọc.

Hoặc nếu $g_{k-1} = 0$ và $g_k \neq 0$ thì

{

- không có bước lọc thứ nhất
- có bước lọc thứ hai trong T_1
- có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo (trong đó $T_1 \neq T$)

}

Hoặc nếu $g_{k-1} \neq 0$ và $g_k = 0$ thì

{

- có bước lọc thứ nhất trong T_1
- không có bước lọc thứ hai
- không có bước lọc thứ ba

}

Hoặc nếu $g_{k-1} \neq 0$ và $g_k \neq 0$ thì chúng ta nhìn vào sự khác biệt của phần nguyên của tần số cơ bản

{

Nếu chênh lệch tuyệt đối giữa phần nguyên của tần số cơ bản trong $k-1$ và k nằm dưới ngưỡng thì

{

- có bước lọc thứ tư trong T_1
- có bước lọc thứ ba trong T (trong đó $T_1 \neq T$)

}

hoặc nếu chênh lệch tuyệt đối giữa phần nguyên của tần số cơ bản trong $k-1$ và k nằm trên ngưỡng thì

- {
 - có bước lọc thứ nhất trong T_1
 - có bước lọc thứ hai trong T_1
 - có bước lọc thứ ba trong khoảng thời gian phụ tiếp theo (tất nhiên chỉ nếu $T_1 \neq T$)
- }
- }

Trong phần 2. thực hiện này, chúng ta cũng có thể bao gồm phần thập phân của tần số cơ bản ngoài việc kiểm tra sự khác biệt giữa phần nguyên của tần số cơ bản.

Fig. 12a thể hiện hệ thống 110 mà có thể triển khai, ví dụ thiết bị mã hóa 80. Hệ thống 110 có thể chứa bộ xử lý 111 và bộ nhớ không tạm thời 112 lưu trữ các lệnh, khi được chạy bởi bộ xử lý 111, có thể khiến bộ xử lý 111 thực hiện ước lượng tham số 113 (ví dụ như trong khối 89), xử lý tín hiệu mã hóa 114 (ví dụ: để triển khai các phần tử 82-86) và dòng bit tạo thành 115 (ví dụ: để triển khai bộ ghi dòng bit 89'). Hệ thống 110 có thể bao gồm đơn vị đầu vào 116, có thể thu được tín hiệu âm thanh (ví dụ: tín hiệu âm thanh 89). Do đó, bộ xử lý 111 có thể thực hiện các quy trình để có được phép biểu diễn được mã hóa (ví dụ: tại định dạng 11a) của tín hiệu âm thanh. Phép biểu diễn được mã hóa này có thể được cung cấp cho các đơn vị bên ngoài bằng cách sử dụng đơn vị đầu ra 117. Đơn vị đầu ra 117 có thể bao gồm, ví dụ, đơn vị truyền thông để liên lạc với các thiết bị bên ngoài (ví dụ: sử dụng giao tiếp không dây, chẳng hạn như Bluetooth) và/hoặc không gian lưu trữ bên ngoài. Bộ xử lý 111 có thể lưu phép biểu diễn được mã hóa của tín hiệu âm thanh trong không gian lưu trữ cục bộ 118.

Fig. 12 thể hiện hệ thống 120 mà có thể triển khai thiết bị 10, 30, 40, 50, 90 và/hoặc thực hiện phương pháp 60 hoặc 70. Hệ thống 120 có thể chứa bộ xử lý 121 và bộ nhớ không tạm thời 122 lưu trữ các lệnh mà, khi được chạy bởi bộ xử lý 121, có thể khiến bộ xử lý 121 thực hiện đọc dòng bit 123 (ví dụ: để triển khai đầu đọc dòng bit 91'), điều khiển bộ lọc thứ nhất/thứ hai 124 (ví dụ: bất kỳ phần tử 12, 14, 22, 24, 51, 52 nào và/hoặc các hoạt động của bộ lọc 61, 62 và/hoặc các bước của phương pháp 70) và/hoặc điều khiển bộ lọc thứ ba 125 (ví dụ, để triển khai bộ lọc thứ ba 31, 53 và/hoặc các hoạt động

của bộ lọc 63 và/hoặc các bước của phương pháp 70). Hệ thống 120 có thể bao gồm đơn vị đầu vào 126, có thể thu được phép biểu diễn được giải mã của tín hiệu âm thanh (ví dụ: ở dạng 11a). Do đó, bộ xử lý 121 có thể thực hiện các quy trình để lọc tín hiệu (ví dụ: sử dụng bộ lọc LTP). Phép biểu diễn được lọc này có thể được cung cấp cho các đơn vị bên ngoài bằng cách sử dụng đơn vị đầu ra 127. Đơn vị đầu ra 127 có thể bao gồm, ví dụ, đơn vị truyền thông để liên lạc với các thiết bị bên ngoài (ví dụ: sử dụng giao tiếp không dây, chẳng hạn như Bluetooth) và/hoặc không gian lưu trữ bên ngoài. Bộ xử lý 121 có thể lưu phép biểu diễn được lọc của tín hiệu âm thanh trong không gian lưu trữ cục bộ 128.

Trong các ví dụ, các hệ thống 110 và 120 có thể là cùng thiết bị.

Tùy thuộc vào các yêu cầu thực hiện nhất định, các ví dụ có thể được thực hiện trong phần cứng. Việc triển khai có thể được thực hiện sử dụng vật ghi lưu trữ số, ví dụ, đĩa mềm, đĩa đa năng kỹ thuật số (Digital Versatile Disc-DVD), đĩa Blu-Ray, đĩa compact (Compact Disc-CD), bộ nhớ chỉ đọc (Read-only Memory-ROM), bộ nhớ chỉ đọc khả trình (Programmable Read-only Memory-PROM), bộ nhớ chỉ đọc khả trình có thể xóa được (Erasable and Programmable Read-only Memory-EPROM), bộ nhớ chỉ đọc khả trình có thể xóa được bằng điện tử (Electrically Erasable Programmable Read-Only Memory-EEPROM) hoặc bộ nhớ nhanh flash, có các tín hiệu điều khiển có thể đọc được bằng điện tử được lưu trữ trên đó, mà kết hợp (hoặc có thể kết hợp) với hệ thống máy tính khả trình sao cho phương pháp tương ứng được thực hiện. Do đó, vật ghi lưu trữ số có thể có khả năng đọc được bằng máy tính.

Nói chung, các ví dụ có thể được triển khai như sản phẩm chương trình máy tính với các lệnh chương trình, các lệnh chương trình có tác dụng để thực hiện một trong số các phương pháp khi sản phẩm chương trình máy tính chạy trên máy tính. Các lệnh chương trình có thể, ví dụ, được lưu trữ trên vật ghi có thể đọc được bằng máy.

Các ví dụ khác bao gồm chương trình máy tính để thực hiện một trong số các phương pháp được mô tả tại đây, được lưu trữ trên vật mang có thể đọc được bằng máy. Do đó, nói cách khác, ví dụ về phương pháp là chương trình máy tính có các lệnh chương trình để thực hiện một trong số các phương pháp được mô tả tại đây, khi chương trình máy tính chạy trên máy tính.

Do đó, ví dụ khác của các phương pháp là vật mang dữ liệu (hoặc vật ghi lưu trữ số, hoặc vật ghi có thể đọc được bằng máy tính) gồm có, đã được ghi lại trên đó, chương trình máy tính để thực hiện một trong số các phương pháp được mô tả tại đây. Vật mang dữ liệu, vật ghi lưu trữ số hoặc vật ghi là hữu hình và/hoặc không tạm thời, thay vì tín hiệu vô hình và tạm thời.

Ví dụ khác gồm có bộ xử lý, ví dụ máy tính, hoặc thiết bị logic khả trình thực hiện một trong số các phương pháp được mô tả tại đây.

Ví dụ khác bao gồm máy tính có chương trình máy tính được cài đặt trên đó để thực hiện một trong số các phương pháp được mô tả tại đây.

Ví dụ khác gồm có thiết bị hoặc hệ thống truyền tải (ví dụ, theo cách điện tử hoặc quang học) chương trình máy tính để thực hiện một trong số các phương pháp được mô tả tại đây đến bộ nhận. Bộ nhận có thể là, ví dụ, máy tính, thiết bị di động, thiết bị nhớ hoặc tương tự. Thiết bị hoặc hệ thống có thể, ví dụ, gồm có máy chủ tập tin để chuyển chương trình máy tính đến bộ nhận.

Theo một số ví dụ, thiết bị logic khả trình (ví dụ, mảng cổng khả trình được dạng trường) có thể được sử dụng để thực hiện một số hoặc tất cả các chức năng của các phương pháp được mô tả tại đây. Trong một số ví dụ, mảng cổng khả trình dạng trường có thể kết hợp với bộ vi xử lý để thực hiện một trong số các phương pháp được mô tả tại đây. Thông thường, các phương pháp có thể được thực hiện bởi thiết bị phần cứng thích hợp bất kỳ.

Các ví dụ được mô tả tại trên được minh học cho các nguyên tắc được thảo luận tại trên. Cần hiểu rằng các biến thể và biến đổi của các phương án và các chi tiết được mô tả tại đây sẽ là rõ ràng. Do đó, mục đích của sáng chế chỉ bị giới hạn bởi phạm vi của các điểm yêu cầu bảo hộ sắp đưa ra dưới đây và không bị giới hạn bởi các chi tiết cụ thể được thể hiện qua các phần mô tả và giải thích của các phương án tại đây.

YÊU CẦU BẢO HỘ

1. Hệ thống lọc tín hiệu thông tin đầu vào, được chia thành các khoảng thời gian cập nhật khác nhau, theo các hệ số thay đổi theo các khoảng thời gian cập nhật, để thu được tín hiệu đầu ra được lọc, trong đó tín hiệu thông tin đầu vào là tín hiệu âm thanh, hệ thống bao gồm:

bộ lọc thứ nhất để lọc tín hiệu đầu vào bộ lọc thứ nhất ít nhất tại khoảng thời gian phụ ban đầu trong khoảng thời gian cập nhật hiện tại để xuất ra tín hiệu đầu ra bộ lọc thứ nhất, theo các hệ số liên quan đến khoảng thời gian cập nhật trước đó, bộ lọc thứ nhất được tạo cấu hình để định tỉ lệ các hệ số dọc theo ít nhất là khoảng thời gian phụ ban đầu, trong đó bộ lọc thứ nhất được tạo cấu hình để định tỉ lệ các hệ số liên quan đến khoảng thời gian cập nhật trước đó bởi hệ số định tỉ lệ thứ nhất thay đổi về 0 hoặc giá trị gần 0, dọc theo ít nhất là khoảng thời gian phụ ban đầu; và

bộ lọc thứ hai để lọc tín hiệu đầu vào bộ lọc thứ hai, tại khoảng thời gian phụ ban đầu, theo các hệ số liên quan đến khoảng thời gian cập nhật hiện tại để xuất ra tín hiệu đầu ra bộ lọc thứ hai, bộ lọc thứ hai được tạo cấu hình để định tỉ lệ các hệ số dọc theo ít nhất là khoảng thời gian phụ ban đầu, trong đó bộ lọc thứ hai được tạo cấu hình để định tỉ lệ các tham số liên quan đến khoảng thời gian cập nhật hiện tại bởi hệ số định tỉ lệ thứ hai thay đổi từ 0 hoặc giá trị gần 0, về giá trị khác 0 hoặc giá trị xa 0 hơn so với giá trị gần 0, dọc theo ít nhất khoảng thời gian phụ ban đầu,

trong đó tín hiệu đầu vào bộ lọc thứ nhất là tín hiệu thông tin đầu vào, tín hiệu đầu ra bộ lọc thứ nhất là tín hiệu đầu vào bộ lọc thứ hai, và tín hiệu đầu ra được lọc là tín hiệu đầu ra bộ lọc thứ hai.

2. Hệ thống theo điểm 1, hệ thống còn bao gồm:

bộ lọc thứ ba để lọc tín hiệu thông tin đầu vào tại khoảng thời gian phụ tiếp theo trong khoảng thời gian cập nhật hiện tại sau khoảng thời gian phụ ban đầu theo các hệ số liên quan đến khoảng thời gian cập nhật hiện tại.

3. Hệ thống theo điểm 2, hệ thống còn bao gồm:

bộ chọn được tạo cấu hình để kiểm tra xem khoảng cách giữa tham số cho khoảng thời gian cập nhật hiện tại và tham số cho khoảng thời gian cập nhật trước đó là 0 hay

nằm trong ngưỡng thứ nhất, để lọc tín hiệu thông tin đầu vào ít nhất trong khoảng thời gian phụ ban đầu bằng cách sử dụng bộ lọc thứ ba, trong đó tham số cho khoảng thời gian cập nhật hiện tại và tham số cho khoảng thời gian cập nhật trước đó được lấy từ một trong các thông tin điều hoà, thông tin khuếch đại, thông tin độ trễ tần số cơ bản, phần nguyên của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào và phần phân số của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào.

4. Hệ thống theo điểm 2, trong đó bộ lọc thứ ba được tạo cấu hình, dọc theo khoảng thời gian tiếp theo để duy trì các hệ số bộ lọc không đổi.

5. Hệ thống theo điểm 2, trong đó bộ lọc thứ ba hoạt động như bộ lọc trước dự báo dài hạn (long term prediction-LTP), bộ lọc sau LTP, bộ lọc trước LTP điều hoà, hoặc bộ lọc sau LTP điều hoà.

6. Hệ thống theo điểm 2, hệ thống còn được tạo cấu hình để kiểm tra độ khuếch đại g_k của khoảng thời gian cập nhật thứ k hiện tại và độ khuếch đại g_{k-1} của khoảng thời gian cập nhật thứ $(k-1)$ trước đó, sao cho:

- nếu $g_{k-1}=0$ và $g_k=0$, thì bước lọc tại các bộ lọc thứ nhất, thứ hai và thứ ba được khử kích hoạt; và/hoặc

- nếu $g_{k-1}=0$ và $g_k \neq 0$, thì

- bước lọc tại bộ lọc thứ nhất được khử kích hoạt;
- có bước lọc tại bộ lọc thứ hai trong ít nhất khoảng thời gian phụ ban đầu;
- có bước lọc tại bộ lọc thứ ba trong khoảng thời gian phụ tiếp theo; và/hoặc

- nếu $g_{k-1} \neq 0$ và $g_k=0$, thì

- có bước lọc tại bộ lọc thứ nhất trong ít nhất khoảng thời gian phụ ban đầu;
- bước lọc tại bộ lọc thứ hai được khử kích hoạt;
- bước lọc tại bộ lọc thứ ba được khử kích hoạt; và/hoặc

- nếu $g_{k-1} \neq 0$ và $g_k \neq 0$, thì sự khác biệt của các phần nguyên và phần thập phân của độ trễ tần số cơ bản được kiểm tra, sao cho:

- nếu phần nguyên và phần thập phân của độ trễ tần số cơ bản trong khoảng thời gian cập nhật thứ k hiện tại và trong khoảng thời gian cập nhật thứ $(k-1)$ trước đó là như nhau, thì:

- bước lọc tại các bộ lọc thứ nhất và bộ lọc thứ hai được khử kích hoạt;

- có bước lọc tại bộ lọc thứ ba dọc theo toàn bộ khoảng thời gian cập nhật hiện tại;

- hoặc nếu có sự khác biệt về phần nguyên hoặc phần thập phân của độ trễ tần số cơ bản:

- có bước lọc tại bộ lọc thứ nhất trong ít nhất khoảng thời gian phụ ban đầu;

- có bước lọc tại bộ lọc thứ hai trong ít nhất khoảng thời gian phụ ban đầu;

- có bước lọc tại bộ lọc thứ ba trong khoảng thời gian phụ tiếp theo.

7. Hệ thống theo điểm 1, trong đó ít nhất một trong số bộ lọc thứ nhất và thứ hai hoạt động như bộ lọc trước dự báo dài hạn (long term prediction, - LTP), bộ lọc sau LTP, bộ lọc trước LTP điều hòa, bộ lọc sau LTP điều hòa.

8. Hệ thống theo điểm 1, trong đó ít nhất một trong số bộ lọc thứ nhất và thứ hai có hàm chuyển bao gồm tử số và mẫu số, trong đó tử số bao gồm giá trị độ khuếch đại được biểu thị bằng thông tin độ khuếch đại, và trong đó mẫu số bao gồm phần nguyên của độ trễ tần số cơ bản được biểu thị bằng thông tin độ trễ tần số cơ bản và bộ lọc đa chạm phụ thuộc vào phần thập phân của độ trễ tần số cơ bản.

9. Hệ thống theo điểm 1, trong đó:

các hệ số của bộ lọc thứ nhất và bộ lọc thứ hai được suy ra các tham số được chọn trong số thông tin điều hòa, thông tin độ khuếch đại, thông tin độ trễ tần số cơ bản,

phần nguyên của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào và/hoặc phần thập phân của độ trễ tần số cơ bản của tín hiệu thông tin đầu vào.

10. Hệ thống theo điểm 1, trong đó:

bộ lọc thứ nhất và bộ lọc thứ hai được chọn trong số bộ lọc mã hóa dự báo tuyến tính (linear predictive coding-LPC), bộ lọc đáp ứng xung vô hạn (infinite impulse response-IIR) và/hoặc bộ lọc đáp ứng xung hữu hạn (finite impulse response-FIR).

11. Hệ thống theo điểm 1, trong đó:

hệ số định tỉ lệ thứ nhất và hệ số định tỉ lệ thứ hai là các giá trị không âm bổ sung cho nhau với giá trị lớn hơn 0.

12. Hệ thống theo điểm 1, trong đó:

hệ số định tỉ lệ thứ nhất thay đổi về 0 về cực trị cuối cùng của ít nhất là khoảng thời gian phụ ban đầu, và/hoặc

hệ số định tỉ lệ thứ hai thay đổi từ 0 hoặc giá trị gần 0, từ cực trị ban đầu của khoảng thời gian cập nhật hiện tại về giá trị khác không hoặc giá trị xa 0 hơn so với giá trị gần 0.

13. Hệ thống theo điểm 1, trong đó:

bộ lọc thứ tư được tạo cấu hình để lọc tín hiệu thông tin đầu vào, ít nhất là tại khoảng thời gian phụ ban đầu, sử dụng hệ số thu được bằng cách nội suy các tham số liên quan đến khoảng thời gian cập nhật hiện tại và các tham số liên quan đến khoảng thời gian cập nhật trước đó.

14. Hệ thống theo điểm 13, trong đó bộ lọc thứ tư hoạt động như bộ lọc trước dự báo dài hạn (long term prediction-LTP), bộ lọc sau LTP, bộ lọc trước LTP điều hòa, hoặc bộ lọc sau LTP điều hòa.

15. Hệ thống theo điểm 13, còn bao gồm:

bộ chọn được tạo cấu hình để kiểm tra xem khoảng cách giữa các tham số cho khoảng thời gian cập nhật hiện tại có nằm trong ngưỡng thứ hai hay không, để lọc tín hiệu thông tin đầu vào ít nhất trong khoảng thời gian phụ ban đầu bằng cách sử dụng bộ lọc thứ tư.

16. Hệ thống theo điểm 15, còn được tạo cấu hình để chủ động đặt ngưỡng thứ hai trên cơ sở các giá trị liên quan đến tín hiệu thông tin đầu vào.

17. Hệ thống theo điểm 16, còn được tạo cấu hình để:

thiết lập ngưỡng thứ hai là ngưỡng khoảng cách độ trễ tần số cơ bản bằng với mức tối thiểu giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó,

để sử dụng bộ lọc thứ tư khi khoảng cách giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó nhỏ hơn ngưỡng khoảng cách độ trễ tần số cơ bản; và/hoặc

để sử dụng các bộ lọc thứ nhất và thứ hai khi khoảng cách giữa phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật hiện tại và phần nguyên của độ trễ tần số cơ bản tại khoảng thời gian cập nhật trước đó lớn hơn ngưỡng khoảng cách độ trễ tần số cơ bản.

18. Hệ thống theo điểm 16, còn được tạo cấu hình để:

sử dụng điều kiện liên quan đến độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó,

để sử dụng bộ lọc thứ tư khi cả độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó khác với 0 và/hoặc

để sử dụng bộ lọc thứ nhất và thứ hai khi ít nhất một trong số các độ khuếch đại của tín hiệu tại khoảng thời gian cập nhật hiện tại và tại khoảng thời gian trước đó là không.

19. Hệ thống theo điểm 1, trong đó bộ lọc thứ nhất là để cung cấp tín hiệu đầu ra bộ lọc thứ nhất là:

$$y'[n] = x[n] + s_{k-1}[n] \left(\sum_{i=0}^P b_{k-1,i} x[n-i] - \sum_{j=1}^Q a_{k-1,j} y'[n-j] \right),$$

$$kT \leq n < kT + T_l$$

trong đó hệ số định tỉ lệ $s_{k-1}[n]$ thay đổi về giá trị gần 0 khi n tăng,

và bộ lọc thứ hai là để cung cấp tín hiệu đầu ra bộ lọc thứ hai là:

$$y[n] = y'[n] + s_k[n] \left(\sum_{i=0}^P b_{k,i} y'[n-i] - \sum_{j=1}^Q a_{k,j} y[n-j] \right),$$

$$kT \leq n < kT + T_l$$

trong đó hệ số định tỉ lệ $s_k[n]$ thay đổi từ giá trị gần 0 về giá trị khác 0 khi n tăng, trong đó T là khoảng thời gian cập nhật thứ k hiện tại, T_l là khoảng thời gian phụ ban đầu, n là khoảng khác, $x[n]$ là tín hiệu thông tin đầu vào, $b_{k-1,i}$ và $a_{k-1,j}$ là các hệ số liên quan đến khoảng thời gian cập nhật thứ $(k-1)$ trước đó, $a_{k,j}$ và $b_{k,i}$ là các hệ số liên quan đến khoảng thời gian cập nhật thứ k hiện tại, và P và Q liên quan đến loại bộ lọc.

20. Hệ thống theo điểm 1, trong đó bộ lọc thứ nhất được tạo cấu hình để cung cấp tín hiệu đầu ra bộ lọc thứ nhất dưới dạng

$$\widehat{x_{ltpf}}'(n) = \hat{x}(n) - \left(1 - \frac{n}{\frac{N_F}{4}} \right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x_{ltpf}}' \left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k \right) \right]$$

$$for\ n = 0.. \frac{N_F}{4}$$

và bộ lọc thứ hai được tạo cấu hình để cung cấp tín hiệu đầu ra đã lọc ở dạng

$$\widehat{x_{ltpf}}(n) = \widehat{x_{ltpf}}'(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x_{ltpf}}'(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x_{ltpf}}' \left(n - p_{int} + \frac{L_{den}}{2} - k \right) \right]$$

$$for\ n = 0.. \frac{N_F}{4}$$

trong đó $\frac{N_F}{4}$ là chiều dài của khoảng thời gian phụ ban đầu, $\hat{x}(n)$ là tín hiệu thông tin đầu vào, $\widehat{x_{ltpf}}'$ là tín hiệu đầu ra bộ lọc thứ nhất, $\widehat{x_{ltpf}}(n)$ là tín hiệu đầu ra đã lọc, n là khoảng khác, p_{int}^{mem} và p_{fr}^{mem} tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khoảng thời gian cập nhật trước đó, p_{int} và p_{fr} tương ứng dựa trên phần nguyên và phần thập phân của độ trễ tần số cơ bản liên quan đến khoảng thời gian cập nhật hiện tại, $c_{num}(k)$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật hiện tại, $c_{den}(k, p_{fr})$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật được xác định và dựa trên phần thập phân của tần số cơ bản, $c_{num}^{mem}(k)$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật trước đó,

$c_{den}^{mem}(k, p_{fr}^{mem})$ là hệ số dựa trên giá trị độ khuếch đại cho khoảng thời gian cập nhật trước đó và dựa trên phân thập phân của tần số cơ bản, L_{den} và L_{num} được cố định và/hoặc dựa trên tốc độ lấy mẫu của tín hiệu thông tin đầu vào.

21. Hệ thống theo điểm 1, trong đó độ dài thời gian của khoảng thời gian phụ ban đầu nằm trong khoảng từ 5% đến 40% của độ dài thời gian của khoảng thời gian cập nhật hiện tại.

22. Hệ thống theo điểm 1, bao gồm phía bộ mã hóa và phía bộ giải mã, trong đó ít nhất một trong số bộ lọc thứ nhất và thứ hai nằm tại phía bộ giải mã.

23. Hệ thống theo điểm 1, bao gồm phía bộ mã hóa và phía bộ giải mã, trong đó ít nhất một trong số bộ lọc thứ nhất và thứ hai nằm tại phía bộ mã hóa.

24. Hệ thống theo điểm 9, bao gồm phía bộ mã hóa bao gồm:

bộ ước lượng tham số được tạo cấu hình để ước lượng các tham số cho ít nhất một trong số bộ lọc thứ nhất và thứ hai.

25. Hệ thống theo điểm 1, còn bao gồm bộ chuyển đổi để chuyển đổi phép biểu diễn thứ nhất của tín hiệu thông tin đầu vào thành phép biểu diễn thứ hai của tín hiệu thông tin đầu vào.

26. Hệ thống theo điểm 1, trong đó ít nhất khoảng thời gian phụ là khoảng thời gian cập nhật.

27. Hệ thống theo điểm 1, còn được tạo cấu hình để:

xác định xem bộ lọc thứ nhất và/hoặc thứ hai sẽ hoạt động như bộ lọc nhận dạng; và

trong trường hợp xác định, bỏ qua bộ lọc thứ nhất và/hoặc thứ hai.

28. Hệ thống theo điểm 1, trong đó bộ lọc thứ nhất được tạo cấu hình để định tỉ lệ các hệ số dần dần bằng cách làm giảm và/hoặc một cách đơn điệu hoặc đơn điệu nghiêm ngặt các hệ số dọc theo ít nhất là khoảng thời gian phụ ban đầu.

29. Hệ thống theo điểm 1, trong đó bộ lọc thứ hai được tạo cấu hình để định tỉ lệ các hệ số dần dần bằng cách làm giảm và/hoặc một cách đơn điệu hoặc đơn điệu nghiêm ngặt các hệ số dọc theo ít nhất là khoảng thời gian phụ ban đầu.

30. Phương pháp lọc tín hiệu thông tin đầu vào được chia thành các khoảng thời gian cập nhật khác nhau, theo các hệ số liên quan đến các khoảng thời gian cập nhật, để xuất ra tín hiệu đầu ra được lọc, trong đó tín hiệu thông tin đầu vào là tín hiệu âm thanh, phương pháp bao gồm các bước:

thực hiện bước lọc thứ nhất ít nhất tại khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại theo các hệ số liên quan đến các khoảng thời gian cập nhật trước đó, trong đó các hệ số dọc theo ít nhất khoảng thời gian phụ ban đầu được định tỉ lệ bởi hệ số định tỉ lệ thứ nhất thay đổi về 0, hoặc giá trị gần 0, dọc theo ít nhất khoảng thời gian phụ ban đầu; và

thực hiện bước lọc thứ hai ít nhất tại khoảng thời gian phụ ban đầu, theo các hệ số liên quan đến khoảng thời gian cập nhật hiện tại, trong đó các hệ số dọc theo khoảng thời gian phụ ban đầu được định tỉ lệ bởi hệ số định tỉ lệ thứ hai thay đổi từ 0, hoặc giá trị gần 0, về giá trị khác 0, hoặc giá trị xa 0 hơn so với giá trị gần 0, dọc theo ít nhất khoảng thời gian phụ ban đầu,

trong đó bước lọc thứ nhất được thực hiện trên tín hiệu thông tin đầu vào và bước lọc thứ hai được thực hiện trên tín hiệu thu được từ bước lọc thứ nhất.

31. Vật ghi lưu trữ số không tạm thời có chương trình máy tính được lưu trữ trên đó để thực hiện phương pháp lọc tín hiệu âm thanh đầu vào được chia thành các khoảng thời gian cập nhật khác nhau, theo các hệ số liên quan đến các khoảng thời gian cập nhật, để xuất ra tín hiệu đầu ra được lọc, phương pháp bao gồm các bước:

thực hiện bước lọc thứ nhất ít nhất tại khoảng thời gian phụ ban đầu của khoảng thời gian cập nhật hiện tại theo các hệ số liên quan đến các khoảng thời gian cập nhật trước đó, trong đó các hệ số dọc theo ít nhất khoảng thời gian phụ ban đầu được định tỉ lệ bởi hệ số định tỉ lệ thứ nhất thay đổi về 0, hoặc giá trị gần 0, dọc theo ít nhất khoảng thời gian phụ ban đầu; và

thực hiện bước lọc thứ hai ít nhất tại khoảng thời gian phụ ban đầu, theo các hệ số liên quan đến khoảng thời gian cập nhật hiện tại, trong đó các hệ số dọc theo khoảng thời gian phụ ban đầu được định tỉ lệ bởi hệ số định tỉ lệ thứ hai thay đổi từ 0, hoặc giá trị gần 0, về giá trị khác 0, hoặc giá trị xa 0 hơn so với giá trị gần 0, dọc theo ít nhất khoảng thời gian phụ ban đầu,

trong đó bước lọc thứ nhất được thực hiện trên tín hiệu âm thanh đầu vào và bước lọc thứ hai được thực hiện trên tín hiệu được xuất ra bởi bước lọc thứ nhất, khi chương trình máy tính đã nêu chạy trên máy tính.

1/15

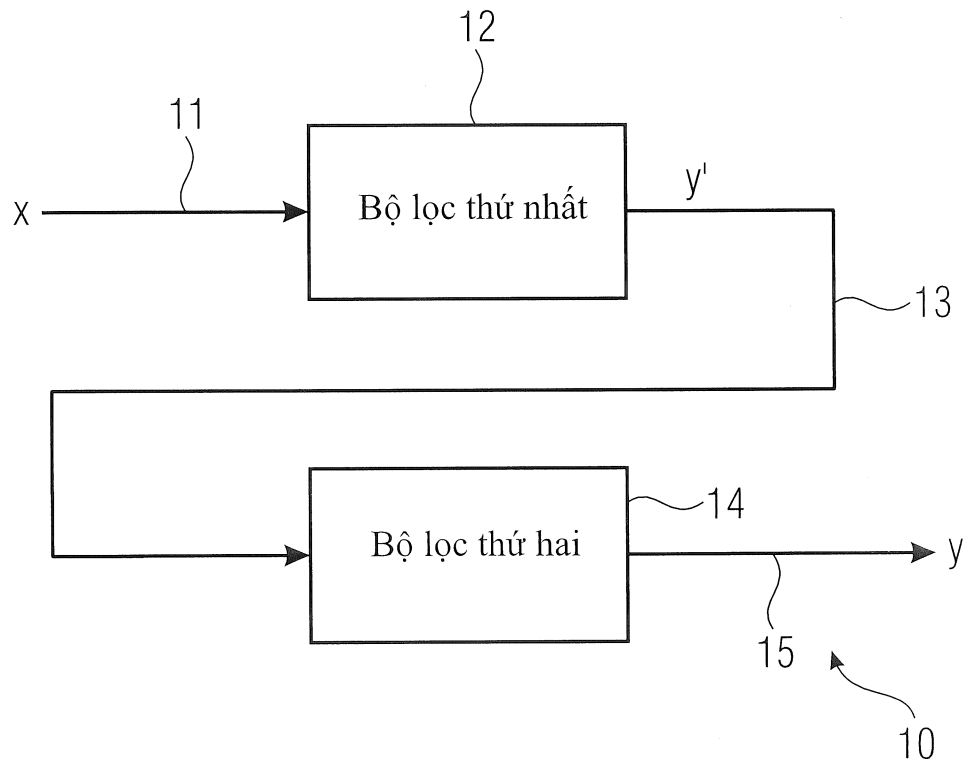


Fig. 1

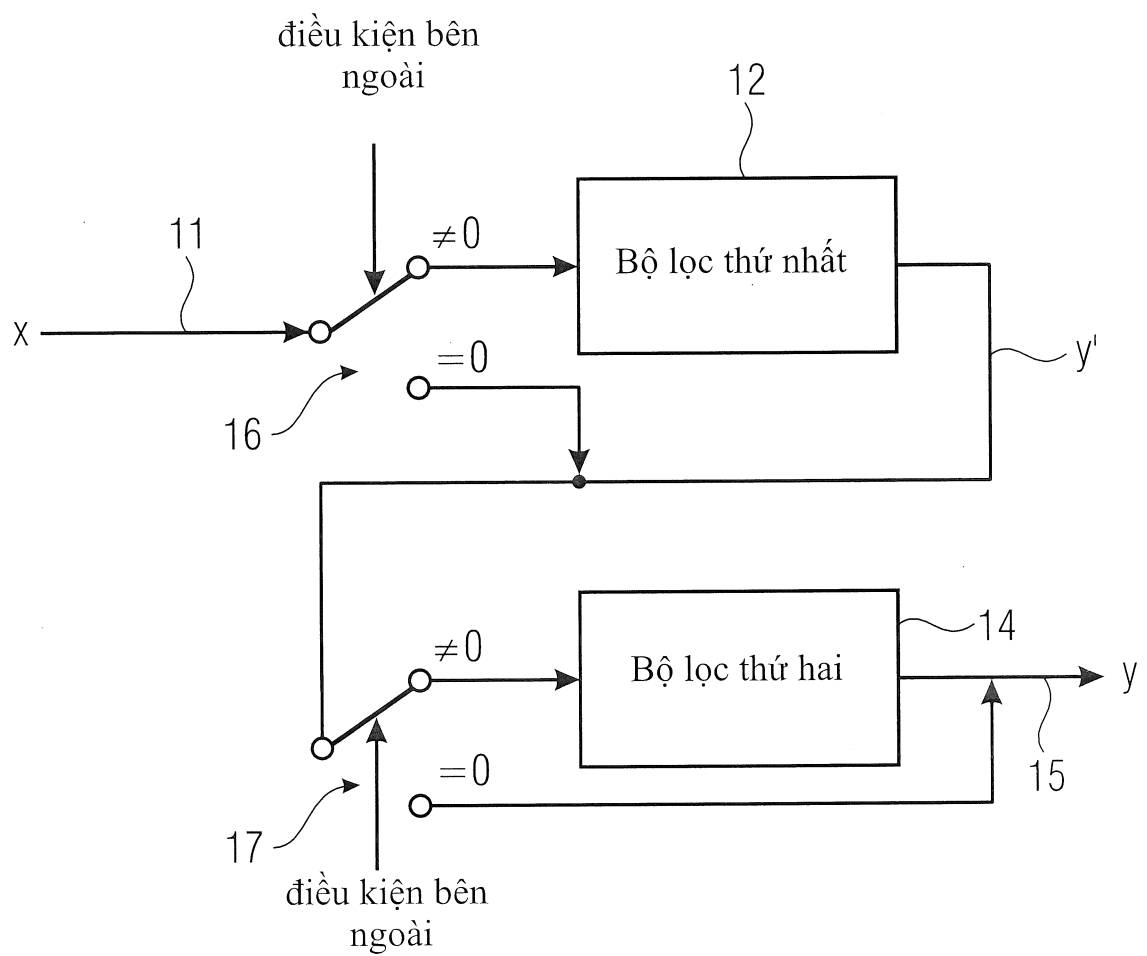


Fig. 1a

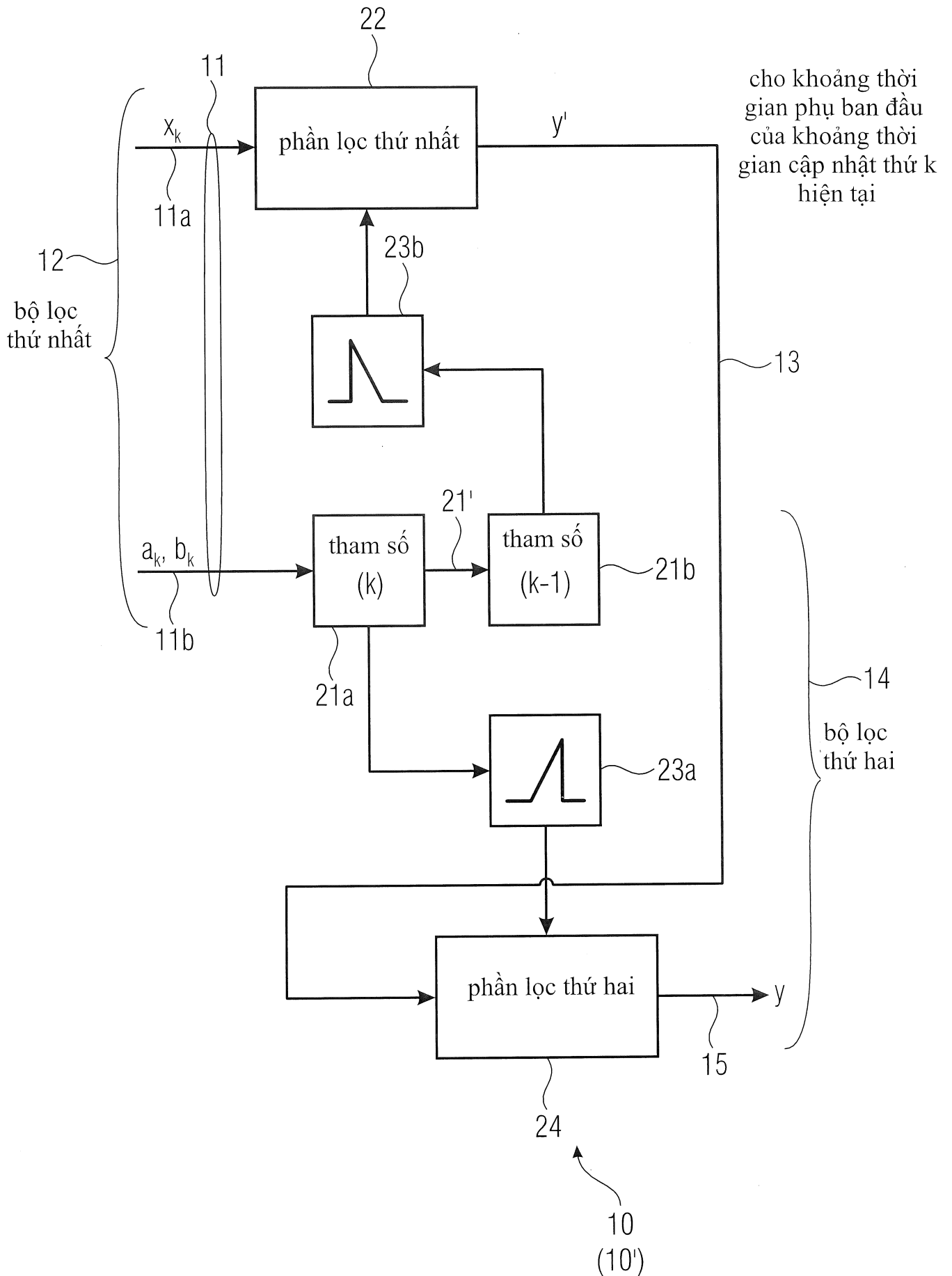


Fig. 2

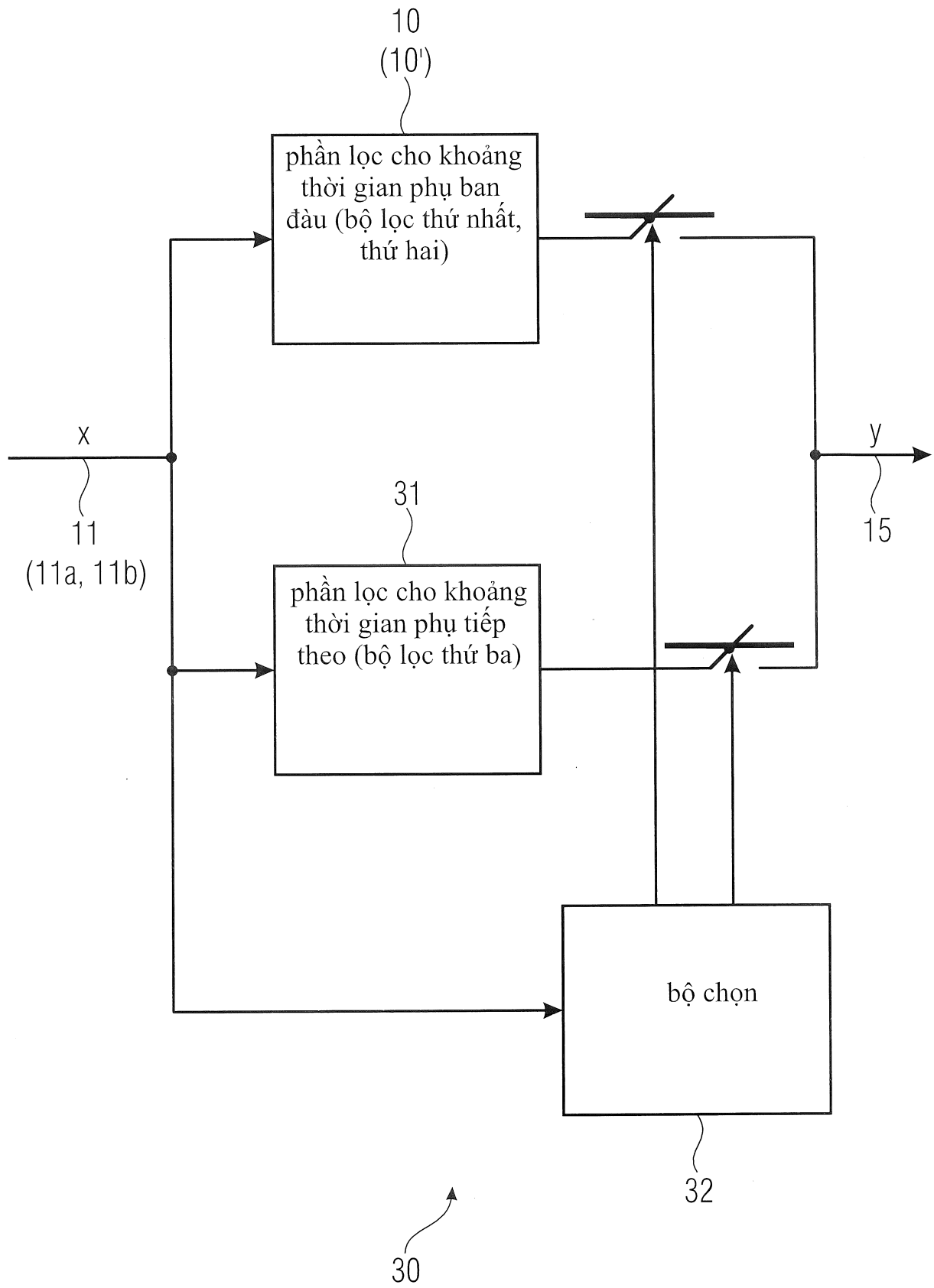


Fig. 3

4/15

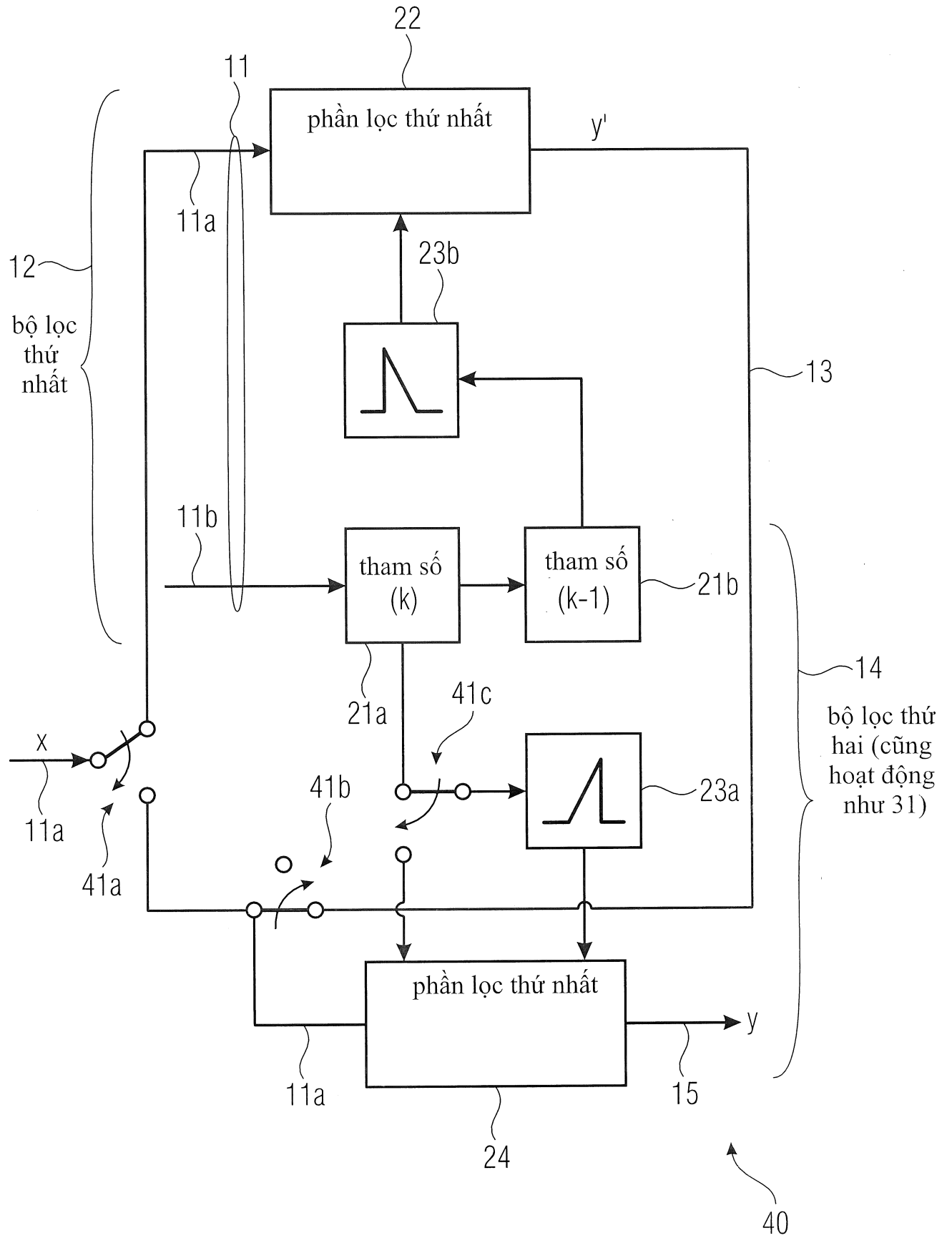


Fig. 4

5/15

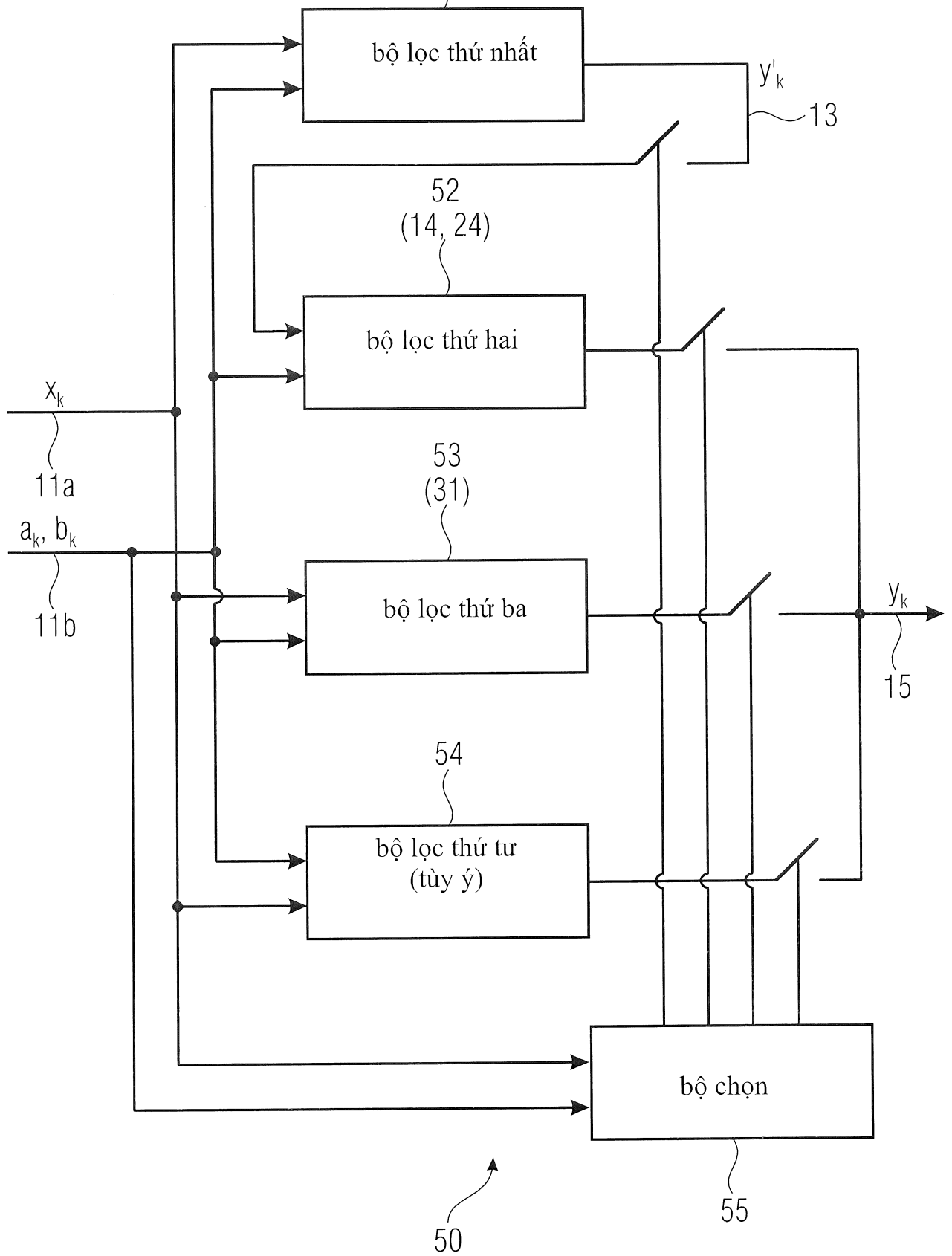
51
(12, 22)

Fig. 5a

6/15

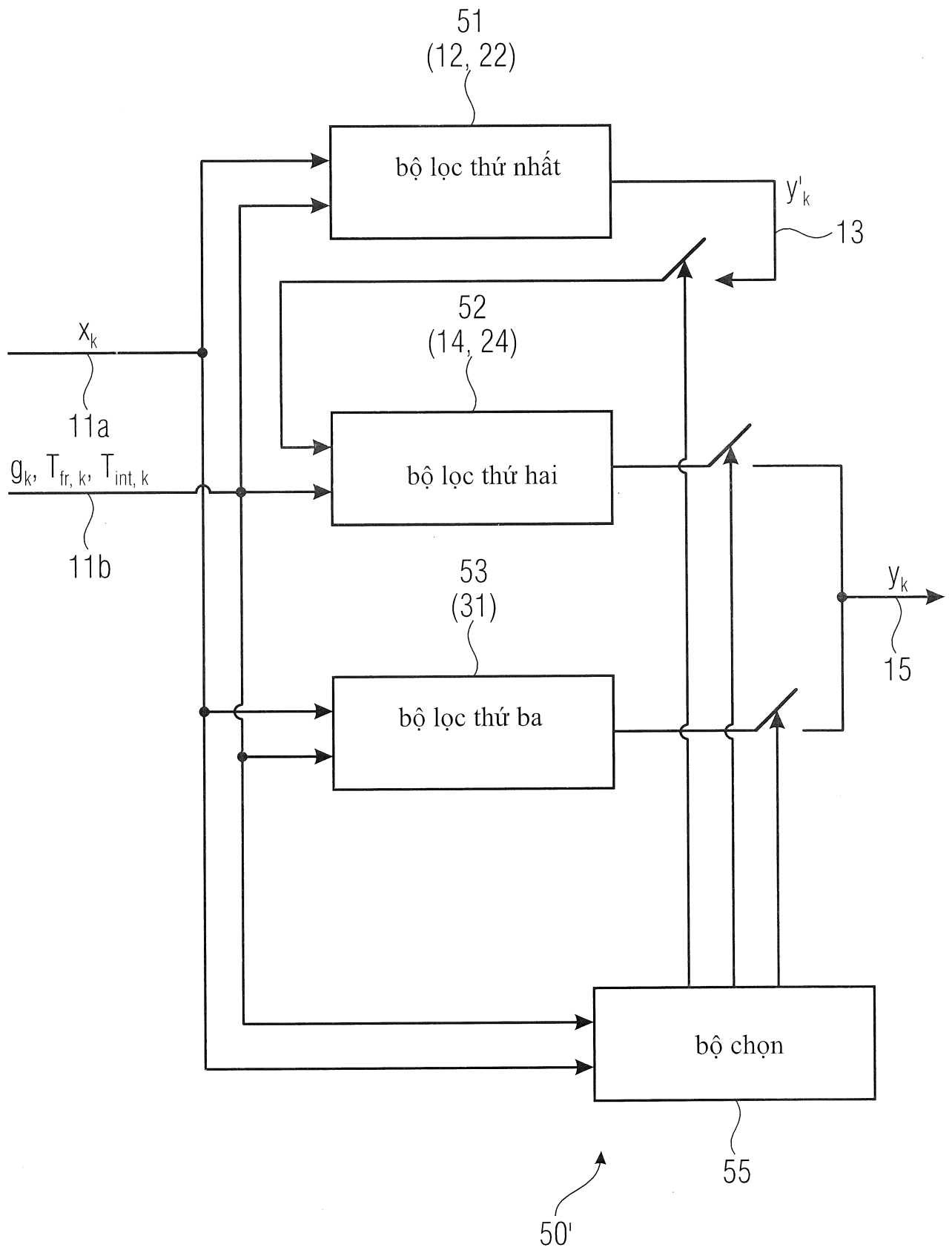


Fig. 5b

UI = khoảng thời gian cập nl
SI = khoảng thời gian phụ

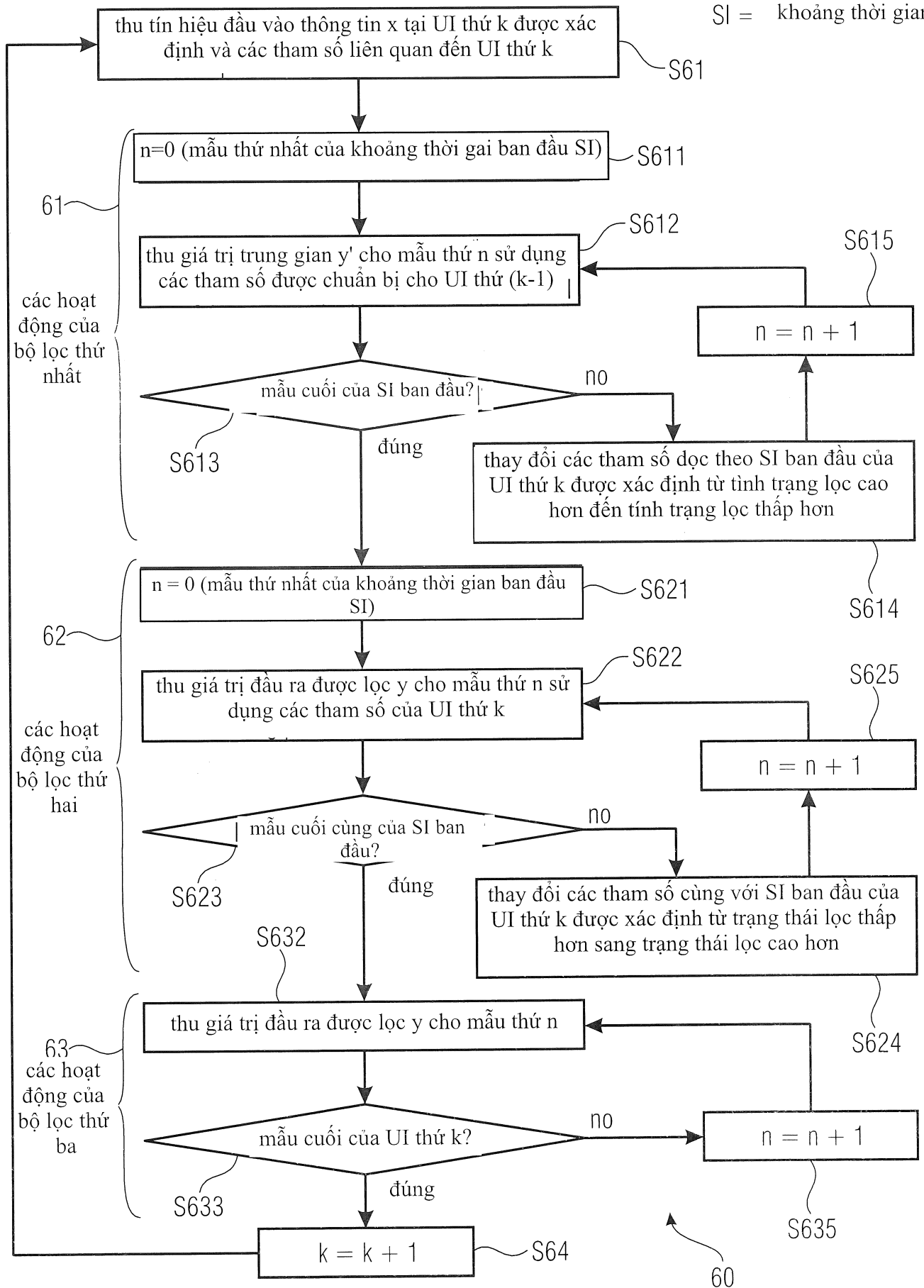


Fig. 6

UI = khoảng thời gian cập n

SI = khoảng thời gian phụ

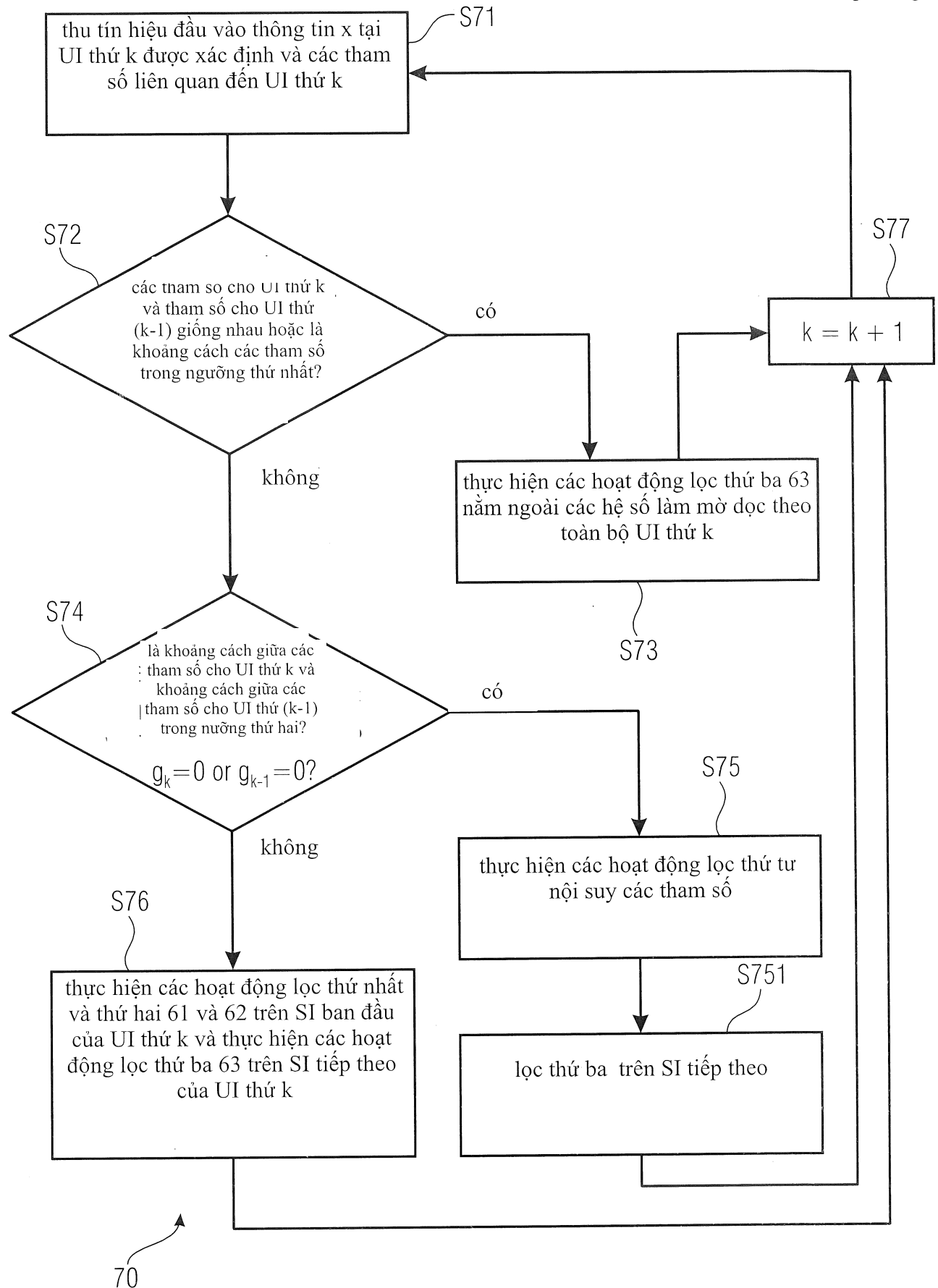


Fig. 7

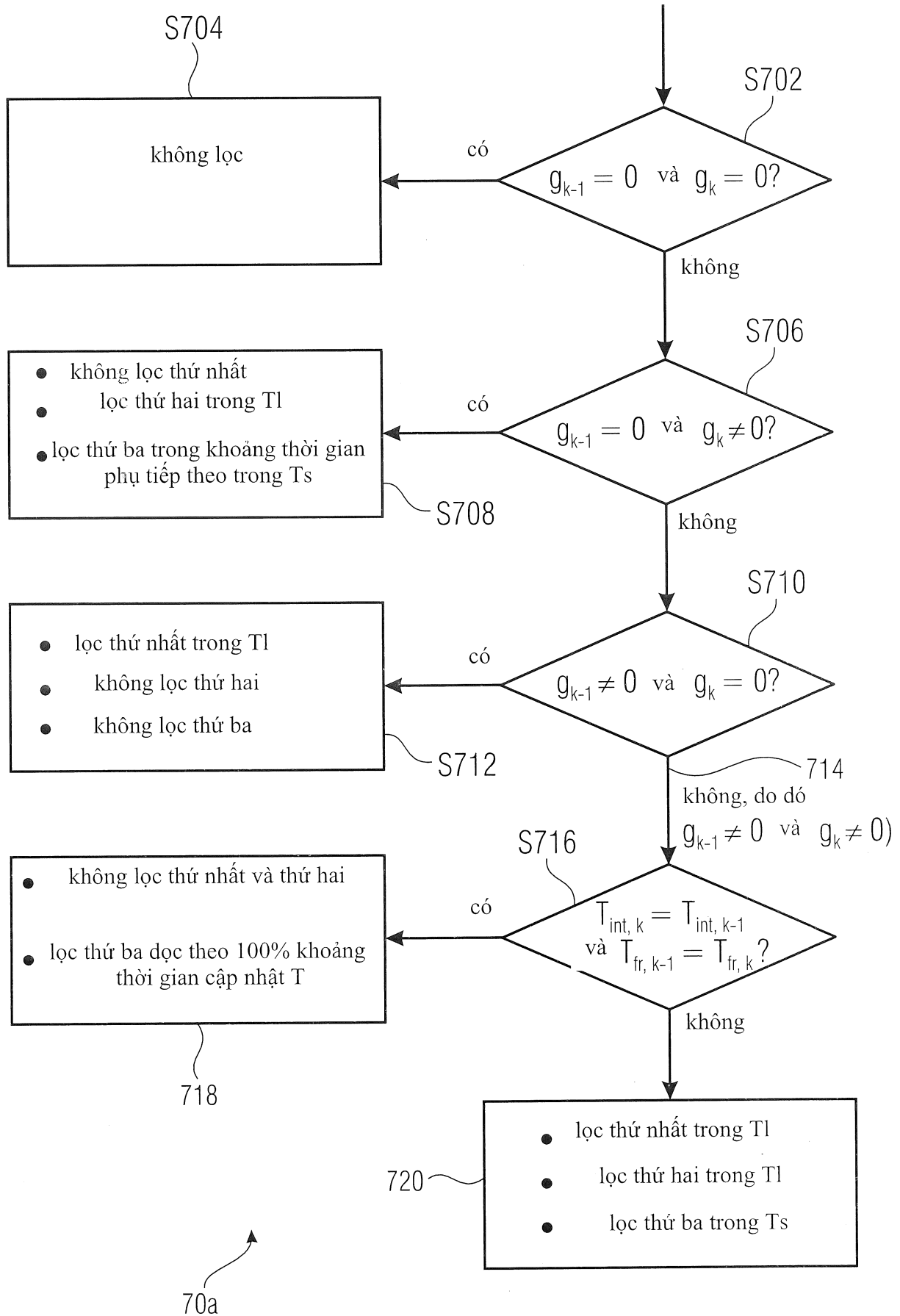


Fig. 7a

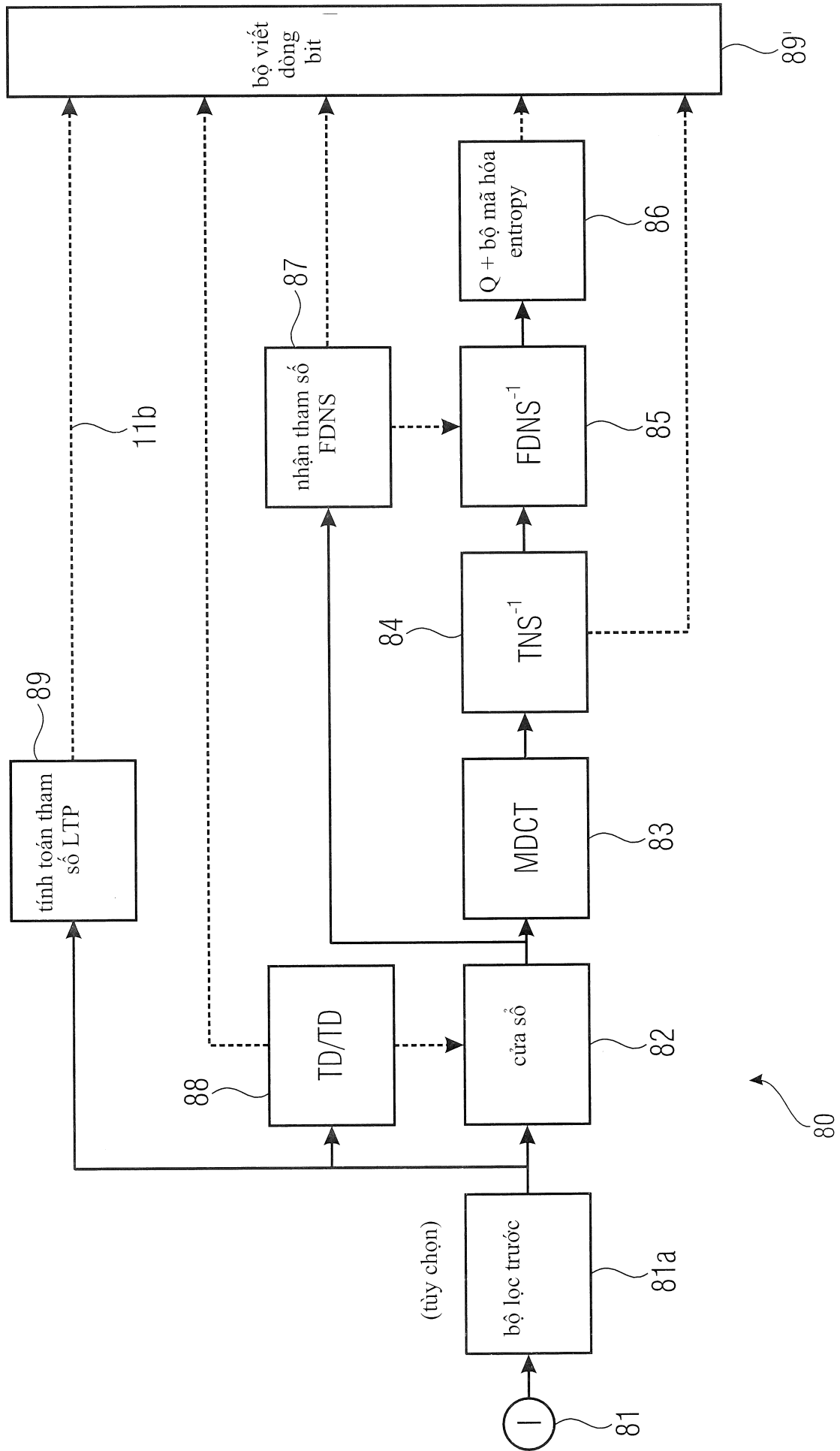


Fig. 8

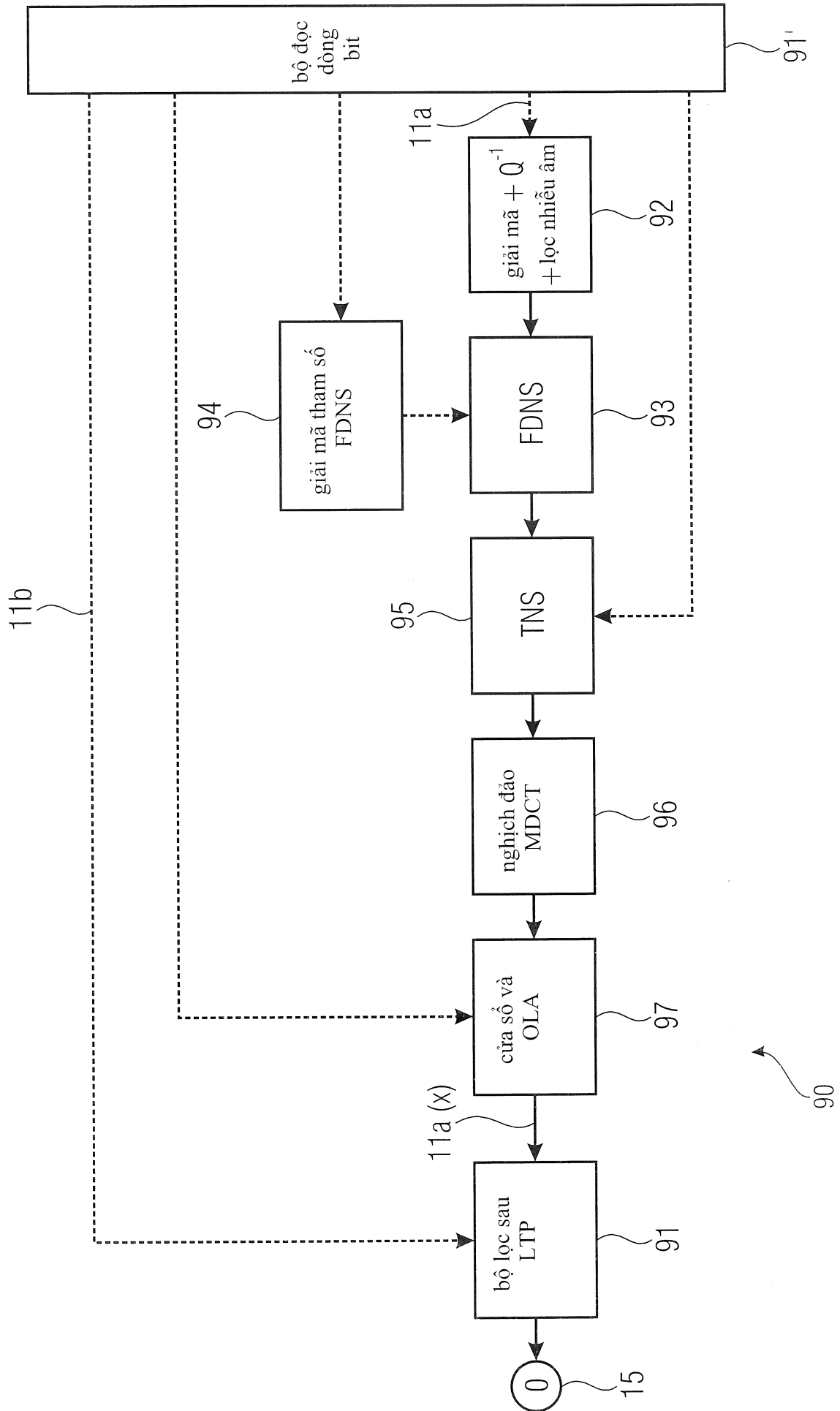


Fig. 9

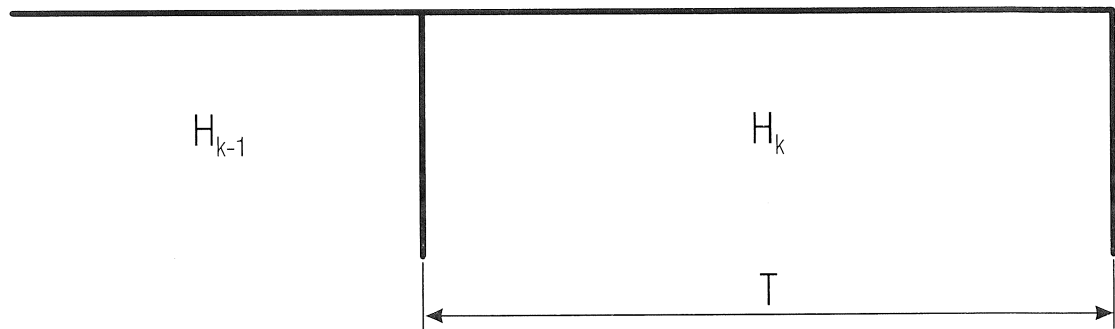


Fig. 10

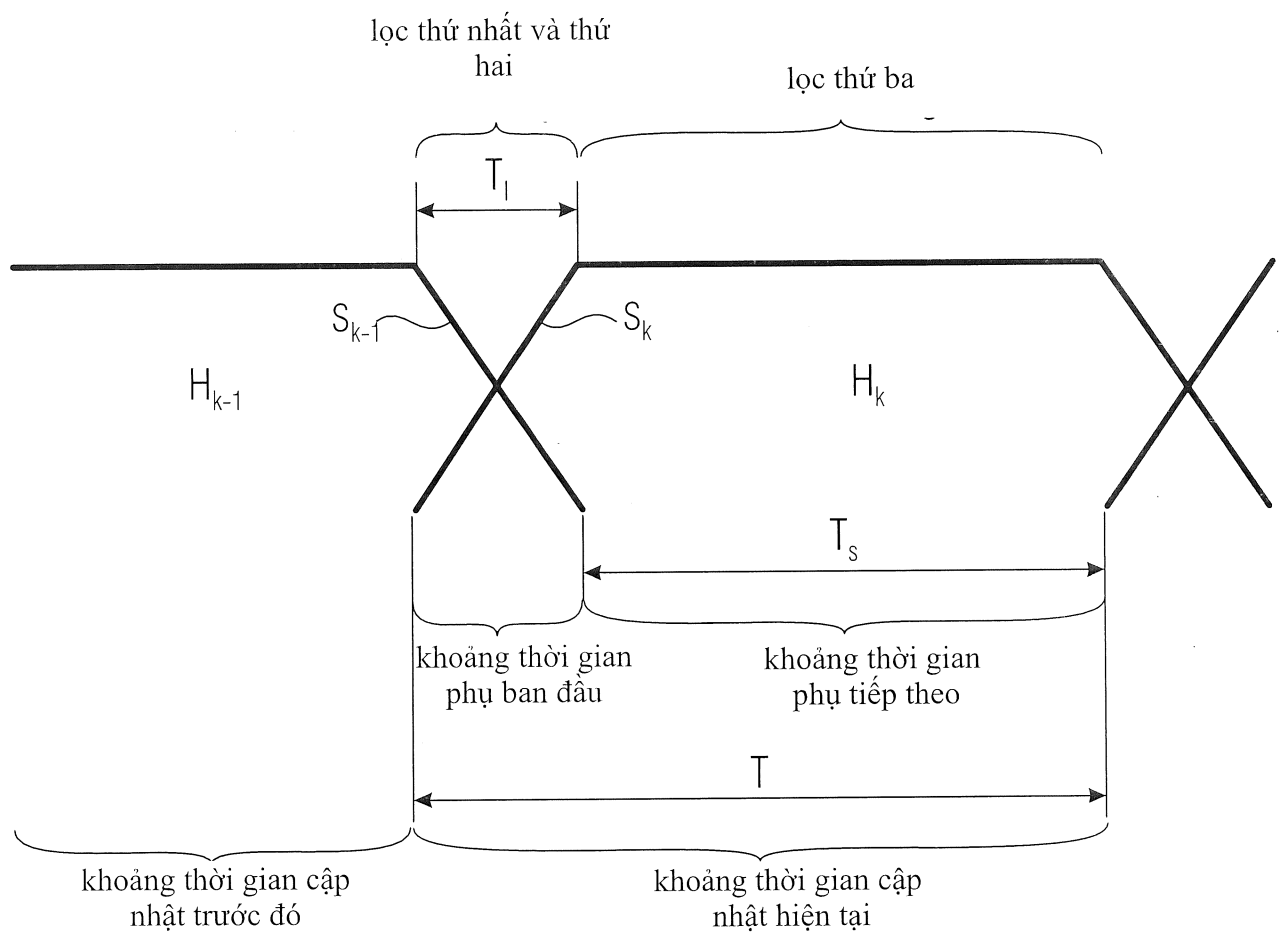


Fig. 11

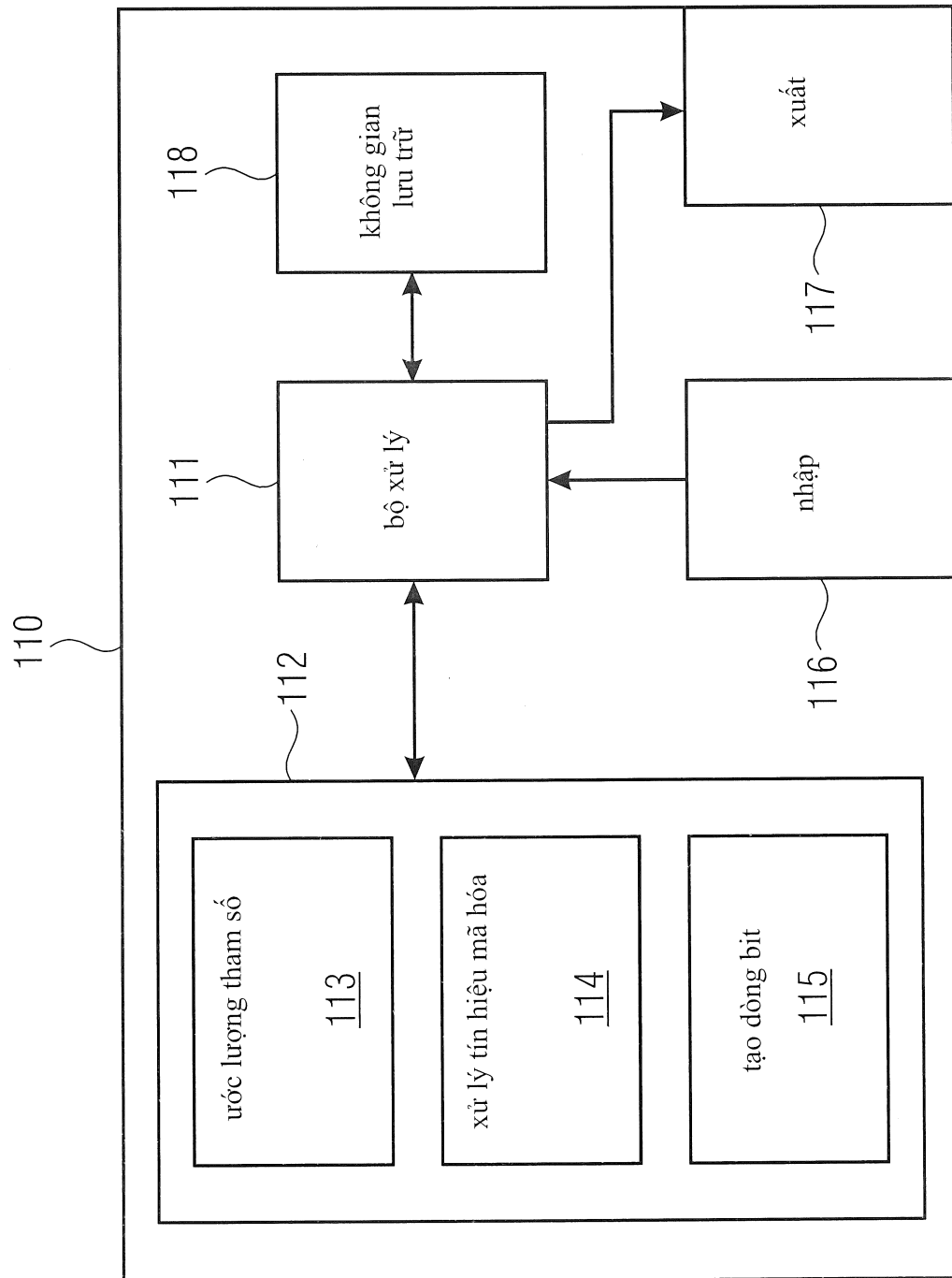


Fig. 12a

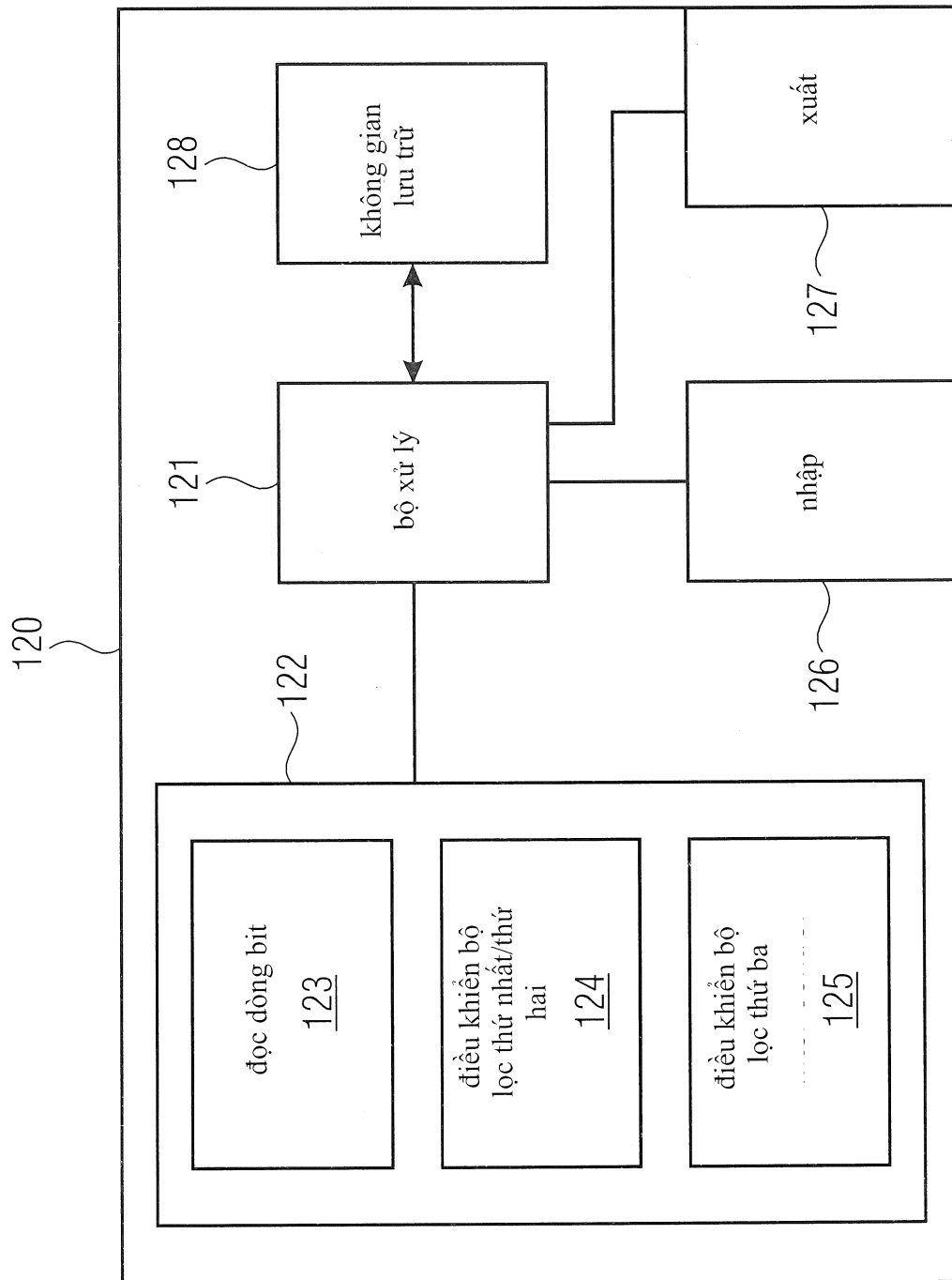


Fig. 12b

130



$$H(z) = \frac{1 - \alpha\beta g B(z, 0)}{1 - \beta g B(z, T_{fr}) z^{-T_{int}}}$$

g : giá trị độ khuếch đại

T_{fr} : phần thập phân của độ trễ tần số cơ bản

T_{int} : phần nguyên của độ trễ tần số cơ bản

$B(z, T_{fr})$: bộ lọc FIR đa chạm phụ thuộc vào phần thập phân

$B(z, 0)$: bộ lọc FIR đa chạm cho phần thập phân số 0

α : giá trị vô hướng thứ nhất $0,6 \leq \alpha \leq 1,0$

β : giá trị vô hướng thứ hai $0,1 \leq \beta \leq 0,5$

Fig. 13