



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

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

(11)



1-0029398

(51)⁷ G10H 5/07; G10H 7/08

(13) B

(21) 1-2015-02887

(22) 07/08/2015

(45) 25/09/2021 402

(43) 27/02/2017 347A

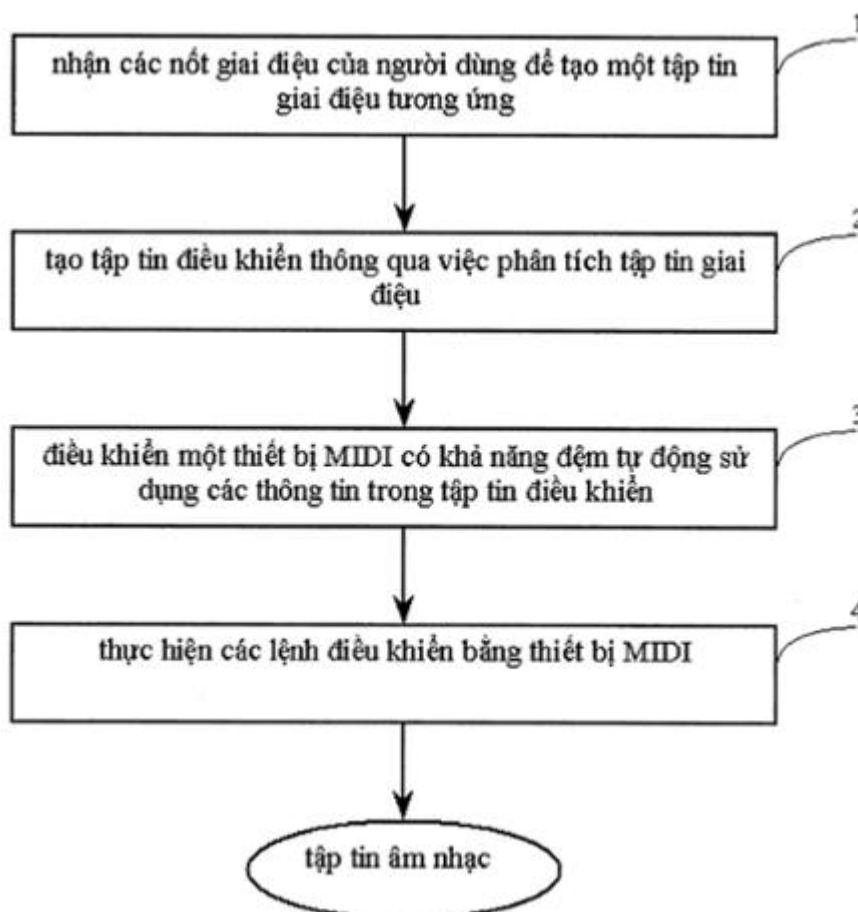
(73) Nguyễn Anh Kiệt (VN)

26 Lý Tự Trọng, phường Bến Nghé, quận 1, thành phố Hồ Chí Minh

(72) Nguyễn Anh Kiệt (VN); Trần Việt Hưng (VN).

(54) PHƯƠNG PHÁP HÒA ÂM TỰ ĐỘNG

(57) Sáng chế đề cập đến phương pháp hòa âm tự động bao gồm các bước: nhận các nốt giai điệu của người dùng để tạo một tập tin giai điệu tương ứng; tạo tập tin điều khiển thông qua việc phân tích tập tin giai điệu; điều khiển một thiết bị MIDI có khả năng đệm tự động sử dụng các thông tin trong tập tin điều khiển để điều khiển một thiết bị MIDI (Musical Instrument Digital Interface - Giao diện kỹ thuật số dành cho nhạc cụ) có khả năng đệm tự động; thực hiện các lệnh điều khiển để tạo ra âm nhạc và ghi lại thành tập tin âm nhạc bởi thiết bị MIDI có khả năng đệm tự động.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến lĩnh vực xử lý âm thanh, cụ thể là đề cập đến phương pháp hòa âm tự động với hỗ trợ của các phần mềm máy tính, cộng với khả năng đệm tự động của các loại đàn điện tử do các hãng khác nhau sản xuất như Yamaha, Korg, Roland,...

Tình trạng kỹ thuật của sáng chế

Về cơ bản, âm nhạc được tạo dựa trên ba yếu tố: Giai điệu, Hòa âm, và Nhịp điệu.

Giai điệu là thành phần cơ bản của âm nhạc, một yếu tố đại diện cho biểu hiện âm nhạc và cảm xúc của con người.

Hợp âm là sự kết hợp cùng lúc của nhiều âm thanh, có cao độ khác nhau.

Hòa âm là phương án phân bố các hợp âm cho từng nhịp trong bài.

Giai điệu là sự kết nối liên tiếp (theo thời gian) của các âm thanh, có cao độ và chiều dài thời gian.

Để cho một chuỗi âm thanh như vậy có ý nghĩa âm nhạc, cần thêm vào một phần nữa, gọi là Nhịp điệu.

Con người sáng tác nhạc bằng cách bày tỏ cảm xúc của mình trong giai điệu rồi hoàn chỉnh bài hát bằng cách kết hợp lời hát với giai điệu. Tuy nhiên, người bình thường không phải là chuyên gia âm nhạc rất khó soạn được hòa âm và soạn được nhịp điệu đệm phù hợp với giai điệu do chính họ sáng tác ra. Do vậy, nhiều nghiên cứu đã được thực hiện về thiết bị tạo ra âm nhạc có thể tự động soạn hòa âm, và tự động tạo ra nhịp điệu đệm phù hợp với giai điệu được sáng tác bởi những người bình thường, giúp họ diễn tả được cảm xúc của mình.

Hiện nay nhiều loại đàn điện tử có khả năng tự động đệm, người dùng có thể chọn một kiểu đệm (nhịp điệu) được thiết kế sẵn trên đàn để đàn tự động đệm. Người dùng chỉ việc dùng tay trái bấm các hợp âm cần thiết và dùng tay phải chơi các nốt giai điệu của bài. Trong khi đàn đang đệm tự động, người dùng sẽ bấm thêm các nút Dồn trống (Fill-in), hoặc Chuyển kiểu (Main), Dẫn nhập

(Intro), hoặc Kết thúc (Ending). Tuy nhiên, để chơi (play) được như vậy, người dùng phải có đôi tay chuyên nghiệp, điều luyện của một nhạc công.

Bản chất kỹ thuật của sáng chế

Giải pháp kỹ thuật theo sáng chế được đề cập đến trong tài liệu này là một phương pháp hòa âm tự động, trong đó các thao tác bằng tay của người dùng như vừa nêu trên được tự động hóa hoàn toàn. Nhờ vậy, một người không có khả năng chơi đàn bằng tay, vì nhiều lý do khác nhau, vẫn có thể dùng các loại đàn đó để tạo ra một bản hòa âm chuyên nghiệp cho một bài hát bất kỳ, bài hát đó có thể là do chính người dùng sáng tác ra.

Phương pháp hòa âm tự động này đã được tác giả thử nghiệm thành công với các loại đàn: Yamaha PSR-1500, PSR-2100; Korg Pa3Xle; Roland Bk9, Bk7m, Bk5, Bk3; Casio LK-50, MZ-2000; cho kết quả rất tốt.

Phương pháp hòa âm tự động bao gồm các bước cụ thể như sau:

1. Nhận các nốt giai điệu của người dùng để tạo một tập tin giai điệu tương ứng. Bước này có thể thực hiện bằng một phần mềm có sẵn, ví dụ “Encore”.

2. Tạo tập tin điều khiển thông qua việc phân tích tập tin giai điệu. Bước này được thực hiện bằng phần mềm “Midi Utility” do tác giả viết.

3. Sử dụng các thông tin trong tập tin điều khiển để điều khiển một thiết bị MIDI (Musical Instrument Digital Interface - Giao diện kỹ thuật số dành cho nhạc cụ) có khả năng đệm tự động. Bước này được thực hiện bằng phần mềm “MidiOut Player” do tác giả viết.

4. Thực hiện các lệnh điều khiển để tạo ra âm nhạc và ghi lại thành tập tin âm nhạc bởi thiết bị MIDI có khả năng đệm tự động. Bước này có thể được thực hiện bằng một đàn điện tử do các hãng khác nhau sản xuất, ví dụ Yamaha, Korg, Roland,...

Mô tả văn tắt hình vẽ

Hình 1 là sơ đồ khối thể hiện các bước trong phương pháp hòa âm tự động theo sáng chế này.

Hình 2 là minh họa kết quả của giải thuật/các bước lập trình để tự động soạn hòa âm.

Hình 3 là minh họa nội dung của tập tin điều khiển.

Mô tả chi tiết sáng chế

Phương án cơ bản thực hiện sáng chế theo hình 1, trong đó chi tiết các bước thực hiện phương pháp hòa âm tự động cụ thể như sau:

1. Nhận các nốt giai điệu của người dùng để tạo ra một tập tin giai điệu tương ứng.

Tại bước 1 (hình 1), có thể sử dụng một phần mềm nhập nốt nhạc bất kỳ hiện có trên thế giới, ví dụ “Encore”. Người dùng sẽ nhập các nốt nhạc chính của giai điệu từ một bản nhạc in trên giấy hoặc do chính người dùng sáng tác ra. Xuất bản nhạc đó ra tập tin loại MIDI có đuôi là “.mid”

Trong đó, “Encore” được tạo cấu hình để thực hiện việc:

nhận các nốt nhạc và lời hát do người dùng nhập vào, hoặc lấy các nốt nhạc từ một tệp MIDI có sẵn;

in ra giấy (văn bản) các nốt nhạc và lời hát đó trên khuôn nhạc;

xuất ra tệp tin MIDI (Input.mid) chứa các nốt nhạc do người dùng đã nhập vào.

2. Tạo tập tin điều khiển thông qua việc phân tích tập tin giai điệu.

Tại bước 2 (hình 1), sử dụng phần mềm “Midi Utility” do tác giả viết, đã được giải “Triển Vọng” tại cuộc thi Nhân Tài Đất Việt năm 2005 và giải Ba, cuộc thi Châu Á - Thái Bình Dương (APICTA), tại Singapore, năm 2007. Phần mềm “Midi Utility” thực hiện nhiều khâu quan trọng để xuất ra một tập dữ liệu điều khiển dạng MIDI (*.mid). Điểm cơ bản khác biệt của phương pháp này là khả năng xử lý thông minh của phần mềm “Midi Utility”.

Cụ thể, phần mềm “Midi Utility” được tạo cấu hình để thực hiện:

lấy các nốt nhạc của giai điệu trong tệp MIDI (Input.mid) do Encore hoặc Studio tạo ra;

tự động xác định giọng (key) của giai điệu;

tự động soạn hòa âm cho tất cả các nhịp;

cho người dùng chọn một /vài kiểu đệm trong các kiểu đã soạn trước, kèm theo;

tạo ra phần đệm tự động gồm nhiều nhạc cụ tham gia;
xuất kết quả đệm tự động ra tệp MIDI (Output.mid).

Nếu người dùng muốn dùng các kiểu đệm trên đàn Organ, thì Midi Utility tạo ra tệp dạng Output.mid chỉ có các nốt giai điệu, và các lệnh điều khiển đàn (hợp âm, dẫn nhập, dòn trống, kết thúc, ...).

Có 2 giải thuật xử lý thông minh đã được lập trình sẵn trong phần mềm này, cụ thể như sau:

2a. Giải thuật phát hiện giọng (còn gọi là điệu thức/khóa) của giai điệu đầu vào.

Một giai điệu đầu vào do người dùng tự sáng tác, hoặc là văn bản nhạc in trên giấy, thì có thể dễ dàng biết được giọng chính của nó nhờ số dấu thăng hoặc giáng trên hóa biểu và nốt kết thúc. Tuy nhiên, phần mềm có khả năng biết được giọng của giai điệu mà không cần có hóa biểu và nốt kết thúc là một phần mềm có giải thuật xử lý thông minh, một dạng của Trí tuệ nhân tạo.

Phần mềm “Midi Utility” được cài đặt một hàm gọi là “Hàm Heuristic”, do tác giả thiết kế. Hàm này được thiết kế dựa trên các lý thuyết chung nhất trong âm nhạc cộng với kinh nghiệm thực tế của tác giả. Trong đó người dùng có thể tự mình khai báo giọng của giai điệu, hoặc lựa chọn kết quả tìm giọng do phần mềm đưa ra. Hiện nay, không có phần mềm nào trên thế giới giải quyết được việc đó tốt như “Midi Utility”.

Giải thuật xác định giọng cho một ca khúc bất kỳ đã là luận văn tốt nghiệp đã được bảo vệ thành công của một Thạc sỹ công nghệ thông tin, trường Đại học Khoa học Tự nhiên, thành phố Hồ Chí Minh, năm 2003. Luận văn này đã hoàn thành với sự hướng dẫn chi tiết của tác giả của giải pháp kỹ thuật “Công nghệ hòa âm thông minh”.

Chi tiết giải thuật xác định giọng của một giai điệu đầu vào gồm các bước sau:

Lập 12 bảng dữ liệu là 12 “Tập giá trị của các nốt thuộc Giọng” của tất cả 12 giọng trưởng (điệu thức trưởng) và các giọng Thứ tương ứng:

01-Đô Trưởng: 60, 62, 64, 65, 67, 69, 71; Giọng Thứ tương ứng:

La Thứ;

02-Đô# Trưởng: 61, 63, 65, 66, 68, 70, 60; Giọng Thứ tương ứng:
La#/Sib Thứ;

03-Rê Trưởng: 62, 64, 66, 67, 69, 71, 61; Giọng Thứ tương ứng: Si
Thứ;

04-Rê# Trưởng: 63, 65, 67, 68, 70, 60, 62; Giọng Thứ tương ứng:
Đô Thứ;

05-Mi Trưởng: 64, 66, 68, 69, 71, 61, 63; Giọng Thứ tương ứng:
Do#/Reb Thứ;

06-Fa Trưởng: 65, 67, 69, 70, 60, 62, 64; Giọng Thứ tương ứng:
Rê Thứ;

07-Fa# Trưởng: 66, 68, 70, 71, 61, 63, 65; Giọng Thứ tương ứng:
Re#/Mib Thứ;

08-Sol Trưởng: 67, 69, 71, 60, 62, 64, 66; Giọng Thứ tương ứng:
Mi Thứ;

09-Sol# Trưởng: 68, 70, 60, 61, 63, 65, 67; Giọng Thứ tương ứng:
Fa Thứ;

10-La Trưởng: 69, 71, 61, 62, 64, 66, 68; Giọng Thứ tương ứng:
Fa#/Solb Thứ;

11-La# Trưởng: 70, 60, 62, 63, 65, 67, 69; Giọng Thứ tương ứng:
Sol Thứ;

12-Si Trưởng: 71, 61, 63, 64, 66, 68, 70; Giọng Thứ tương ứng:
Sol#/Lab Thứ;

Lấy “Tập các nốt của giai điệu đầu vào” từ dữ liệu đầu vào có định dạng
MIDI do người dùng đã nhập.

So sánh xem “Tập các nốt của giai điệu đầu vào” giống tập dữ liệu nào
trong 12 tập đã chuẩn bị trước nêu trên để xác định được giai điệu đầu vào là
“Giọng Trưởng” hay “Giọng Thứ tương ứng” nào.

Tìm “Các nốt được sử dụng nhiều” trong “Tập các nốt của giai điệu đầu
vào”.

So sánh “Các nốt được sử dụng nhiều” với “Hợp âm gốc của giọng
Trưởng”, và “Hợp âm gốc của giọng Thứ tương ứng”. Nếu “Các nốt được sử

dụng nhiều” giống “Hợp âm gốc của giọng Trưởng”, thì kết luận giai điệu đầu vào có giọng là Giọng Trưởng (là 1 trong 12 giọng Trưởng, theo 12 bảng đã lập nêu trên). Nếu “Các nốt được sử dụng nhiều” giống “Hợp âm gốc của giọng Thứ tương ứng”, thì kết luận giai điệu đầu vào có giọng là Giọng Thứ tương ứng (là 1 trong 12 giọng Thứ tương ứng theo 12 bảng đã lập nêu trên).

2b. Giải thuật tự động soạn hòa âm cho mọi ca khúc.

Giọng của một giai điệu là chìa khóa để soạn hòa âm cho giai điệu đó. Từ các nốt của giai điệu trong các nhịp và giọng đã biết được tại bước 2a, phần mềm “Midi Utility” có thể tìm được các hợp âm thích hợp để dùng trong các nhịp khi hòa âm cho toàn bài thành một ca khúc hoàn chỉnh.

Người dùng có thể tự mình khai báo các hợp âm cho từng nhịp, hoặc chỉnh sửa kết quả do phần mềm đưa ra. Hiện nay, không có phần mềm nào trên thế giới giải quyết được việc đó tốt như “Midi Utility”. Điều này khác hoàn toàn với việc mà nhiều phần mềm khác làm được là phát hiện hợp âm đang dùng trong một bản hòa âm có sẵn ở dạng MIDI. Từ các nốt hòa âm dễ dàng suy ra được hợp âm đang dùng.

Tuy nhiên, một giai điệu đầu vào thường chỉ có đơn lẻ từng nốt một nối tiếp nhau chứ không có nhiều nốt cùng một lúc như các bản hòa âm. Thậm chí, trong một ca khúc có thể có nhiều nhịp không có nốt nào, ví dụ trong các nhịp dẫn nhập, trong các nhịp giữa hai lần hát, trong các nhịp kết. Do vậy, trong các trường hợp vừa nêu trên không thể dựa vào các nốt hòa âm để tìm hợp âm cần dùng.

Chỉ những người có hiểu biết rất tốt về lý thuyết nhạc, hoặc dày dạn kinh nghiệm thực tế như nhạc sỹ, nhạc công, giáo viên giảng dạy âm nhạc,... mới có khả năng soạn hòa âm cho một giai điệu. Những người khác sẽ cần đến sự giúp đỡ thông minh của phần mềm “Midi Utility”. Ngay cả với những nhạc sỹ, nhạc công, thì “Midi Utility” cũng giúp họ tiết kiệm được rất nhiều thời gian và công sức, họ chỉ cần chỉnh sửa thêm kết quả soạn hòa âm do phần mềm “Midi Utility” đưa ra.

Phần mềm “Midi Utility” được tác giả cài đặt một giải thuật thông minh giải quyết rất nhanh việc soạn hợp âm cho từng nhịp trong bài, kể cả các nhịp

không có một nốt nhạc nào, như dẫn nhập, kết thúc, khoảng nghỉ giữa các câu nhạc, đoạn nhạc. Thời gian cần cho xử lý chỉ khoảng một vài giây.

Giải thuật tự động soạn hòa âm cho một ca khúc đã là “Luận văn thạc sỹ công nghệ thông tin” của chính tác giả của giải pháp kỹ thuật theo sáng chế này, được bảo vệ thành công năm 2000 tại trường Đại Học Bách Khoa thành phố Hồ Chí Minh.

Chi tiết giải thuật (đã đơn giản đi, cho dễ hiểu) tự động soạn hòa âm gồm các bước sau:

Trong giải thuật này, giọng của giai điệu đầu vào đã được xác định. Ta sẽ làm tiếp các bước sau để thực hiện việc tự động soạn hòa âm hoàn chỉnh cho toàn bài.

Lập bảng dữ liệu chứa 03 hợp âm chính của tất cả 12 giọng Trưởng và 03 hợp âm chính của tất cả 12 “giọng Thứ tương ứng”. “Hợp âm chính của giọng” là các hợp âm bậc I, IV, và V7 của giọng đó.

Lấy 03 hợp âm chính của giọng Trưởng, là các hợp âm bậc I, IV, và V7 của giọng trưởng đã xác định được.

Lấy 03 hợp âm chính của “giọng Thứ tương ứng”, là các hợp âm bậc I, IV, và V7 của “giọng Thứ tương ứng” đã xác định được.

Chia tất cả các nhịp có trong bài (từ nhịp thứ 1 đến nhịp cuối cùng) thành 02 “Nửa Nhịp”. Ta sẽ tìm cho mỗi nửa nhịp này 01 hợp âm tối ưu phù hợp với giai điệu đầu vào và tuân theo đúng các nguyên tắc cơ bản trong lý thuyết hòa âm.

Thiết kế một “Hàm đánh giá độ tối ưu” khi chuyển hợp âm (là một hàm Heuristic). Hàm này chính là hàm được tác giả thiết kế khéo léo để đưa được các nguyên tắc chung nhất trong lý thuyết hòa âm, và kinh nghiệm thực tế khi hòa âm cho các ca khúc, vào phương pháp tự động soạn hòa âm trình bày trong đơn sáng chế này.

Lấy tất cả các “Hợp âm có thể dùng được” cho tất cả các nửa nhịp. “Hợp âm có thể dùng được” tại một nửa nhịp nào đó là các hợp âm thỏa mãn các điều kiện sau:

- Là 01 trong 06 hợp âm chính của giọng Trưởng, và “giọng Thứ tương ứng” (đã xác định),
- Có “Nốt quan trọng” trong nửa nhịp đó. “Nốt quan trọng” là nốt có tổng trường độ lớn nhất trong nửa nhịp đang xét.
- Có các nốt khác trong nửa nhịp đó,
- Tại nhịp cuối cùng của bài, “Hợp âm có thể dùng được” chỉ duy nhất là “Hợp âm gốc” của giọng.

Thiết kế một “Hàm đánh giá phương án” (cũng là hàm Heuristic) để loại bỏ bớt các “Phương án xấu”. Do sẽ có rất rất nhiều phương án như vậy, ta không thể xét tất cả các phương án (trong CNTT gọi là “Vết Cạn”) mà phải áp dụng một biện pháp hiệu quả để lọc bỏ các “Phương án xấu”, chỉ xét các “Phương án tốt”. “Hàm đánh giá phương án” sẽ làm việc này.

Bắt đầu từ nửa nhịp thứ 1 (đầu tiên, trong bài), lần lượt xét tất cả các “Phương án tốt” để chuyển hợp âm từ một “Hợp âm có thể dùng được” tại nửa nhịp này sang một “Hợp âm có thể dùng được” của nửa nhịp tiếp theo, cho đến nửa nhịp cuối cùng trong bài. Cộng dồn điểm của mỗi lần chuyển hợp âm như vậy (do “Hàm đánh giá độ tối ưu” tính được), ta sẽ được “Điểm tối ưu” của mỗi phương án. Ghi nhớ điểm tối ưu của phương án đang xét và cách chuyển hợp âm (từ hợp âm nào chuyển sang hợp âm nào) của phương án đó.

Tìm “Phương án tốt nhất” là phương án có “Điểm tối ưu” lớn nhất trong tất cả các “Phương án tốt” đã xác định được nêu trên. Lấy các hợp âm trong từng nửa nhịp, của “Phương án tốt nhất” (đã ghi nhớ) làm phương án hòa âm kết quả cho tất cả các nửa nhịp trong bài.

Trong tập tin điều khiển, do phần mềm “Midi Utility” tạo ra, có các thông tin sau:

- Các nốt giai điệu của người dùng.
- Các hợp âm cho từng nhịp trong bài.
- Các chế độ đệm cho từng nhịp.

Các thông tin này là cần thiết để điều khiển thiết bị MIDI ở bước tiếp sau.

3. Điều khiển một thiết bị MIDI có khả năng đệm tự động bằng cách sử dụng các thông tin trong tập tin điều khiển.

Tại bước này (hình 1), sử dụng phần mềm “MidiOut Player” do tác giả viết để nạp dữ liệu đã được tạo ra ở bước 2 nêu trên. Phần mềm này có các giao diện tương ứng với các loại đàn khác nhau để điều khiển các loại đàn đó.

Trong đó, “MidiOut Player” là phần mềm được tạo cấu hình để thực hiện các công việc sau:

lấy các nốt giai điệu và các lệnh điều khiển đàn do Midi Utility tạo ra khi người dùng muốn dùng các kiểu đệm trên đàn (Output.mid);

cho người dùng chọn loại đàn (Yamaha, Roland, ...);

chuyển các lệnh điều khiển đàn của Midi Utility thành các lệnh điều khiển loại đàn đã chọn;

sắp xếp theo thứ tự thời gian với độ chính xác đến 1 mili giây (ms);

chèn thêm các xung đồng bộ vào giữa các nội dung lấy từ tệp Output.mid đã nói trên;

gửi lần lượt (theo thời gian thực) các nội dung đó đến đàn qua cáp MIDI In/Out hoặc cáp USB.

Phương tiện kết nối vật lý giữa phần mềm (chạy trên một máy tính cá nhân) và thiết bị MIDI là một dây MIDI In/Out, hoặc dây USB nối máy tính với đàn.

Các nốt giai điệu được phần mềm gửi đến đàn trên MIDI kênh 01. Đàn sẽ phát trực tiếp (thời gian thực) các nốt này tương tự như nó phát các nốt nhạc do tay phải của người chơi bấm trên phần bên phải của bàn phím.

Các hợp âm cho từng nhịp trong bài được phần mềm gửi đến đàn trên MIDI kênh 06 ở dạng “Các nốt xác định hợp âm”. Ví dụ, để gửi hợp âm Đô Trưởng (ký hiệu là C Major), phần mềm sẽ gửi bộ ba nốt Đô, Mi, Sol, có cùng chiều dài và cùng thời gian (cùng một lúc). Khi nhận được bộ ba nốt này đàn sẽ nhận biết được hợp âm cần dùng là Đô Trưởng, tương tự như nó nhận biết được hợp âm từ bộ ba nốt do tay trái của người chơi bấm trên phần bên trái của bàn phím.

Các chế độ đệm cho từng nhịp được phần mềm gửi đến đàn trên kênh điều khiển. Kênh và cách thức điều khiển (Giao thức điều khiển) có thể khác nhau với các loại đàn khác nhau do sự khác nhau trong thiết kế của các hãng sản xuất các

loại đàn đó. Với các loại đàn do cùng một hãng sản xuất, ví dụ Yamaha, thì cách thức điều khiển giống nhau. Phương pháp này đã được thử nghiệm thành công trên ba loại đàn cao cấp do ba hãng Yamaha, Korg, và Roland sản xuất, với ba giao thức điều khiển khác nhau. Các chế độ đệm trên đàn, thường có, là các chế độ: Dẫn Nhập (Intro), Kết Thúc (Ending), Dồn Trống (Fill-In), và “Phần Chính” (Main). Mỗi một chế độ đệm đó còn có từ 1 đến 4 phương án (Variation) khác nhau, từ đơn giản đến phức tạp để người dùng có thể tự chọn trong khi đàn đang đệm tự động.

Ngoài các thông tin đã nói trên, phần mềm “MidiOut Player” còn gửi đến đàn các lệnh điều khiển khác:

- Lệnh chọn nhịp điệu (Select Style). Mỗi một giai điệu đầu vào có thể phù hợp với vài kiểu nhịp điệu cài đặt sẵn trên đàn. Người dùng tự mình quyết định chọn nhịp điệu nào phù hợp tốt nhất với giai điệu của họ. Để truy cập nhanh đến nhịp điệu đó, người dùng có thể nhập mã số của nhịp điệu, “MidiOut Player” sẽ nhanh chóng, chính xác, chọn được nhịp điệu đó trên đàn. Tuy nhiên, khả năng này hiện nay chỉ thực hiện được với các đàn cao cấp do hãng Roland sản xuất, ví dụ hộp đệm (Backing Module) Bk7m.

- Lệnh “Bắt Đầu” (Start), để khởi động hệ thống tự động đệm của đàn.

- Lệnh “Dừng” (Stop), để đàn dừng (stop) chế độ tự động đệm.

- Các xung đồng bộ (Clock). Các xung này được gửi đều đặn, liên tục đến đàn với độ chính xác thời gian rất cao, sai số nhỏ hơn 1 mili giây. Các xung đồng bộ này sẽ là các mốc thời gian để đàn tự động đồng bộ tốc độ (Tempo) của bộ đệm theo tốc độ của giai điệu.

Như vậy, phần mềm “MidiOut Player” là một phần mềm làm một việc chưa có phần mềm nào làm được. Các phần mềm hòa âm khác không tận dụng được khả năng tự động tuyệt vời của các loại đàn cao cấp hiện nay bán trên thị trường khắp thế giới. Một số phần mềm, ví dụ SONAR, CAKEWALK, của hãng Roland kèm theo đàn của họ, được sử dụng với mục đích hoàn toàn khác với mục đích của phương pháp trình bày theo chế này, trong đó cái đàn chỉ là phương tiện phát ra âm thanh cao cấp, vì bản thân phần cứng máy tính không thể phát ra được các âm thanh cao cấp đó (âm thanh như nhạc cụ thật: kèn các loại,

sáo các loại, ghi ta các loại,...). Khả năng đệm tự động của đàn không được sử dụng.

Trong phương pháp hòa âm theo sáng chế này, các chức năng của đàn được sử dụng tối đa để hỗ trợ người dùng tạo ra các bản hòa âm cao cấp, chuyên nghiệp. Đàn (thiết bị MIDI) càng cao cấp, chất lượng âm nhạc thu được tại đầu ra cũng càng cao cấp.

4. Thực hiện các lệnh điều khiển để tạo ra âm nhạc và ghi lại thành tập tin âm nhạc bởi thiết bị MIDI có khả năng đệm tự động.

Trong đó, thiết bị MIDI là một loại đàn điện tử cao cấp, có khả năng đệm tự động.

Tại bước 4 (hình 1), có thể sử dụng đàn điện tử cao cấp, có thể là các loại đàn, hoặc thiết bị khác nhau của các hãng Yamaha, Korg, Roland,... Các đàn điện tử cao cấp hiện nay có khả năng rất mạnh trong việc tạo ra âm nhạc:

- Tạo ra được âm thanh cao cấp, như thật, của rất nhiều (hàng ngàn) nhạc cụ.
- Được cài đặt rất nhiều (hơn 500) Nhịp Điệu (còn gọi là Style, Rhythm), do các nhạc sỹ, nhạc công chuyên nghiệp, dày dặn kinh nghiệm thiết kế sẵn.

Người dùng sẽ tự mình chọn một hoặc vài nhạc cụ để chơi phần giai điệu, và một hoặc vài Nhịp Điệu thích hợp với giai điệu của họ để đệm tự động. Các Nhịp Điệu này được cài đặt sẵn trên đàn, hoặc được thêm vào đàn qua giao diện đĩa mềm, USB, hoặc tải xuống từ mạng Internet.

Dưới sự điều khiển của phần mềm “MidiOut Player”, thiết bị MIDI (đàn) sẽ chơi hoàn toàn tự động để tạo ra một bản nhạc chất lượng cao mà không yêu cầu người dùng phải có đôi tay chuyên nghiệp như các nhạc công, cụ thể như sau:

- Khi đàn nhận được lệnh “Bắt Đầu” (Start), nó sẽ khởi động bộ đệm tự động với độ chính xác thời gian rất cao (sai số nhỏ hơn 1 milli giây).
- Khi đàn nhận được các nốt nhạc của giai điệu, nó sẽ phát trực tiếp, theo thời gian thực, các âm thanh của nhạc cụ được chọn tương ứng với các nốt nhạc.
- Khi đàn nhận được các nốt xác định hợp âm, nó sẽ tự động chuyển hợp âm để hòa âm do phần đệm tạo ra phù hợp với giai điệu tại thời điểm hiện tại.

- Khi đàn nhận được lệnh Dẫn Nhập, và phương án (variation) cụ thể, nó sẽ chơi phần dẫn nhập của nhịp điệu đã chọn theo phương án đó (1,2,3,4 hay A,B,C,D, tùy loại đàn).

- Khi đàn nhận được lệnh Dồn Trống (Fill-In), và phương án (variation) cụ thể, nó sẽ thực hiện chu trình dồn trống của nhịp điệu đã chọn theo phương án đó (1,2,3,4 hay A,B,C,D, tùy loại đàn).

- Khi đàn nhận được lệnh chọn “Phần Chính” (Main), và phương án (variation) cụ thể, nó sẽ thực hiện chu trình chuyển phần chính (main) của nhịp điệu đã chọn theo phương án đó (1,2,3,4 hay A,B,C,D, tùy loại đàn).

- Khi đàn nhận được lệnh “Kết Thúc” (Ending), và phương án (variation) cụ thể, nó sẽ thực hiện chu trình kết thúc của nhịp điệu đã chọn theo phương án đó (1,2,3,4 hay A,B,C,D, tùy loại đàn).

- Trong suốt quá trình đệm tự động theo sự điều khiển này, đàn luôn nhận được các xung đồng bộ là các mốc thời gian. Căn cứ vào các mốc thời gian đó, đàn tự động điều chỉnh tốc độ (Tempo) của bộ đệm. Nhờ vậy bộ đệm sẽ chạy khớp chính xác (về thời gian) với giai điệu.

Các bước nêu trên sẽ được đàn thực hiện theo thời gian thực (real-time). Người dùng không cần, nhưng vẫn có thể thao tác thêm trực tiếp trên đàn để bản nhạc tạo ra thêm phong phú hơn, chuyên nghiệp hơn.

Kết quả đầu ra, của sự kết hợp tất cả các yếu tố này là một bản nhạc phù hợp với giai điệu đầu vào của người dùng.

Kết quả hòa âm nghe được trực tiếp theo thời gian thực, và có thể được ghi lại thành tập tin âm nhạc 5 (hình 1) ở dạng MIDI (*.mid), hoặc dạng Wave (*.wav).

Hiệu quả đạt được khi sử dụng phương pháp này là những người sáng tác, những người yêu thích âm nhạc không chuyên nghiệp có thể dùng các loại đàn tự động để làm các bản hòa âm mẫu cho sáng tác của họ, phục vụ mục đích giới thiệu cho bạn bè, đăng ký quyền tác giả,... Họ có thể sáng tác được các ca khúc, nhưng họ có thể không soạn được hòa âm, và không chơi được các loại đàn điện tử đó. Các nhạc sỹ, nhạc công chuyên nghiệp sẽ tiết kiệm được rất

nhiều thời gian và công sức. Các hãng sản xuất đàn điện tử tự động sẽ có nhiều người dùng hơn vì những người không chơi được đàn vẫn sử dụng được đàn.

Phương án khác thực hiện sáng chế.

Các phần mềm cần dùng trong các bước của phương pháp này có thể được tích hợp ngay trên thiết bị MIDI (đàn) tại khâu sản xuất. Khi đó, không cần phải có máy tính bên ngoài, bản thân thiết bị MIDI có thể thực hiện được việc hoà âm hoàn toàn tự động như đã mô tả.

Sự khác nhau giữa “Midi Utility X5” và “Midi Utility X6”:

X6 có chế độ soạn hòa âm không sử dụng các Nhịp Điệu (style) kèm theo phần mềm Midi Utility do tác giả thiết kế sẵn, mà sử dụng “Các nhịp điệu có trên đàn Organ”. “Các nhịp điệu có trên đàn Organ” rất phong phú, được thiết kế rất chuyên nghiệp, sử dụng các âm thanh nhạc cụ rất cao cấp kèm theo đàn. Mỗi đàn có âm thanh cao cấp riêng của nó (mà các đàn khác có thể không có). Nhịp Điệu của mỗi đàn có điểm độc đáo (khác các đàn khác) sử dụng và phát huy tối đa các âm thanh cao cấp có trên đàn đó;

X6 tạo được “Dữ liệu điều khiển đàn” theo định dạng MIDI;

X6 có thêm phần mềm mới gọi là “X6 Player”, để nạp dữ liệu điều khiển rồi dùng dữ liệu đó để điều khiển đàn;

do các hãng Yamaha, Korg, Roland, Casio thiết kế đàn của họ theo cách khác nhau, nên “X6 Player” phải sử dụng giao thức điều khiển thích hợp cho từng loại đàn.

Tất cả các đặc điểm nêu trên hoàn toàn không có trong X5 và các phiên bản trước.

Ví dụ thực hiện sáng chế

- Bài hát: "Cha Yêu",
- Tự động soạn hòa âm: phần mềm "Make Chord Algorithms V2.46",
- Điều khiển thiết bị MIDI: phần mềm "MidiOut Player V2.04",
- Thiết bị MIDI: Roland Bk-7m,
- Kiểu đệm tự động: Pop
- Kết quả tạo âm nhạc: <https://youtu.be/ffczZLWr0bI>

Giải thuật/các bước lập trình:

Tạo tập tin điều khiển thông qua việc phân tích tập tin giai điệu.

- Lấy các nốt giai điệu trong tập tin giai điệu,
- Chia các nốt đó, theo thời gian, thành các nốt trong từng nhịp,
- Trong mỗi nhịp tìm hai Nốt Quan Trọng, một nốt cho nửa đầu, và một nốt cho nửa cuối của nhịp. Nốt Quan Trọng là nốt có tổng chiều dài thời gian lớn nhất trong nửa nhịp đó.
- Lấy các nốt khác cho từng nhịp.
- Căn cứ vào tập các giá trị khác nhau của các nốt giai điệu có trong toàn bài, tìm Key (còn gọi là giọng, điệu thức, âm giai...) của giai điệu.
- Hoặc lấy thông tin về Key do người dùng cung cấp.
- Căn cứ vào: Key, các Nốt Quan Trọng, các nốt khác trong tất cả các nhịp, tự động soạn hòa âm cho từng nhịp trong bài: hợp âm dùng cho từng nhịp, chế độ đệm cho từng nhịp, các thời điểm thực hiện dồn trống (fill-in), các thời điểm chuyển phương án đệm (variation). Việc này được thực hiện bằng các giải thuật: “Lần Ngược”, “Tính Độ Tối Ưu”, và “Điền Hợp Âm”, “Điền Chế Độ Đệm” sẽ nói rõ hơn trong phần sau.
- Hoặc lấy các hợp âm và chế độ đệm do người dùng cung cấp,
- Ghi các thông tin về hợp âm và chế độ đệm đó vào tập tin điều khiển,
- Chép lại tất cả các nốt của giai điệu vào tập tin điều khiển,

Các việc này do phần mềm “Midi Utility” thực hiện. “Midi Utility” có thể nạp một tệp midi (*.mid) bất kỳ, nhưng sau đó nó chuyển đổi sang một dạng cố định để dễ xử lý ở các bước sau. Các Midi Editor khác có thể nạp được các tệp midi của “Midi Utility”. Nhưng sau khi các editor đó ghi lại, thì khuôn dạng có thể khác đi, làm “MidiOut player” từ chối, không nạp, hoặc báo lỗi.

Như vậy tập tin điều khiển là một tệp Midi chuẩn *.mid, theo khuôn dạng cố định của Midi Utility, có các thông tin sau (Hình 3):

- Các nốt giai điệu đầu vào, được giữ nguyên, không thay đổi (1-DiemXua), các nốt được viết thêm sau khi xử lý (2 - Chords). Các nốt này được ghi lại trên các kênh Midi số 1, 2, 3, 4, 5.
- Các nốt xác định hợp âm cho từng nhịp (3 - ChdCmd), được ghi trên kênh Midi số 6,
- Chế độ đệm cho từng nhịp (4 – Ak-Drums), ghi trên kênh Midi số 10, ở dạng mã lệnh của Midi Utility. Mã lệnh được thống nhất theo một cách để “MidiOut player” hiểu được (không phải là các nốt nhạc).
- Người dùng có thể thêm/bớt/sửa nội dung của tập tin điều khiển, bằng một Midi Editor nào đó tương thích tốt, trước khi nạp vào “MidiOut player”.

Điều khiển thiết bị MIDI có khả năng đệm tự động sử dụng các thông tin trong tập tin điều khiển.

- Nạp tập tin điều khiển vào bộ nhớ.

Lấy từ tập tin điều khiển:

- Các thông tin về nốt giai điệu, hoặc nốt viết thêm (kênh, cao độ, cường độ, thời điểm On, thời điểm Off, ...), đưa vào bộ nhớ đệm (buffer) có tốc độ xử lý nhanh, ở dạng “các thông điệp Midi” (Midi Messages), sắp xếp chính xác theo thứ tự thời gian.
- Tương tự, các nốt xác định hợp âm cũng được đưa vào bộ nhớ đệm đó.
- Các mã lệnh điều khiển chế độ đệm do “Midi Utility” tạo ra (cũng là các thông điệp Midi) được chuyển thành các lệnh điều khiển theo qui định của thiết bị MIDI. Các thiết bị MIDI do các hãng khác nhau sản xuất có thể có các qui định khác nhau về các lệnh điều khiển này.

Lấy từ người sử dụng:

- Phương án Intro A, B, C, D (dẫn nhập) và Ending A, B, C, D (kết thúc) theo sự lựa chọn của người dùng vì có thể rất khác nhau trên các

thiết bị MIDI: chiều dài tính bằng số nhịp, cách chơi, các nhạc cụ tham gia,...

- Loại thiết bị MIDI (Yamaha, Korg, Roland,...) để chuyển đổi các lệnh điều khiển theo qui định của nhà sản xuất thiết bị MIDI.

Gửi ra cổng Midi Out của máy tính:

- Sau khi đưa các thông tin nói trên vào bộ nhớ đệm, theo thứ tự thời gian rất chính xác, “MidiOut player” sẽ chèn thêm vào đó các xung định thời gian (clock, cũng là các thông điệp MIDI), để thiết bị MIDI dùng làm xung nhịp, lệnh Start để bắt đầu, lệnh Intro để thực hiện dẫn nhập, và cuối cùng là lệnh Stop, hoặc Ending.
- Khi nhận được lệnh Ending, thiết bị MIDI sẽ chơi phần kết thúc, sau đó tự động dừng. Trong khi thiết bị MIDI chơi phần Ending (có chiều dài thời gian rất khác nhau), mặc dù không còn lệnh nào khác để gửi, “MidiOut player” vẫn gửi các xung nhịp để thiết bị MIDI hoạt động bình thường. Sau khoảng 24 nhịp, nó sẽ gửi lệnh Stop, rồi dừng lại, không gửi thông tin gì khác nữa.
- Trong kỹ thuật Midi, các lệnh (các thông điệp MIDI) là một chuỗi liên tiếp các byte, nối tiếp nhau, trong đó giá trị của mỗi byte có ý nghĩa xác định, theo thiết kế của nhà sản xuất, hoặc theo chuẩn chung mà các thiết bị MIDI tuân theo.
- Khi bộ nhớ đệm (buffer) đã được chuẩn bị xong, “MidiOut player” sẽ gửi nội dung trong bộ nhớ đệm, rất chính xác về thời gian, và rất nhanh, đến thiết bị MIDI, qua cáp kết nối Midi, hoặc cáp USB.

Nhận các lệnh điều khiển từ cổng Midi In của thiết bị MIDI và thực hiện các lệnh đó:

- Thiết bị MIDI phải được chuyển sang chế độ đồng bộ ngoài (external clock) để nhận thông tin truyền trên cáp, và thực hiện những gì mà nó nhận được.

- Người dùng chọn (trên thiết bị) kiểu đệm (style), nhạc cụ dùng cho các kênh 1, 2, 3, 4, 5 sao cho phù hợp với giai điệu đầu vào. Sau đó họ khởi động việc truyền/nhận lệnh qua cáp, bằng cách bấm nút “Play” của MidiOut player.
- Trong khi thiết bị MIDI thực hiện lệnh do “MidiOut player” gửi, người dùng vẫn can thiệp được vào thiết bị MIDI, để chơi thêm các nốt, fill-in, chuyển variation, ...

Những vấn đề khác liên quan:

- Vì tệp tin điều khiển, do “Midi Utility” tạo ra là một tệp midi chuẩn (*.mid), người dùng có thể dùng một “Midi Editor” nào đó, tương thích, để sửa nội dung trước khi nạp vào “MidiOut player”. Ví dụ: có thể tách giai điệu thành các phần khác nhau, để cho các nhạc cụ khác nhau chơi, hoặc họ viết thêm vào đó các nốt. Họ có thể thêm bớt chiều dài đoạn đầu để khớp với chiều dài Intro trên thiết bị MIDI,...
- Các lệnh midi (Midi Messages) truyền trên cáp, từ MidiOut player đến thiết bị MIDI có thể tham khảo trên các link sau:

<https://www.Midi.org/articles-old/about-midi-part-3-midi-messages>

<http://DigitalSoundAndMusic.com/6-1-5-a-closer-look-at-midi-messages/>

Điều kiện để thực hiện được

- “MidiOut player” phải hiểu được nội dung trong tệp tin điều khiển,
- “MidiOut player” phải biết các thông tin cần thiết để điều khiển thiết bị MIDI, do hãng sản xuất công bố.
- Thiết bị MIDI phải có khả năng nhận xung đồng bộ (external clock) và các lệnh điều khiển từ bên ngoài, qua cáp, và thực hiện các lệnh đó. Các đàn organ cao cấp của hãng Yamaha, Roland, Korg đều có khả năng này. Nhưng mã lệnh và kênh điều khiển thì không giống

nhau. Các đàn organ này hiện nay vẫn điều khiển được nhau. “MidiOut player” sẽ tương tự như một cái đàn đó, điều khiển một cái đàn khác. Nhưng “MidiOut player” lấy thông tin từ tệp tin điều khiển, không cần người dùng phải thao tác bằng tay.

Các thuật toán liên quan đến phần mềm “MidiOut player”

- Không có các thuật toán gì đặc biệt liên quan đến “MidiOut player”. Những người có kỹ năng lập trình tốt, có hiểu biết chi tiết về kỹ thuật Midi là có thể làm được.

Các thuật toán / bước lập trình liên quan đến phần mềm “Midi Utility”

- Thuật toán tìm “Nốt Quan Trọng”,
- Thuật toán “Lần Ngược” để tìm phương án soạn hòa âm tối ưu,
- Hàm “Đánh Giá Độ Tối Ưu” dùng trong thuật toán “Lần Ngược”,
- Thuật toán “Điền Hợp Âm”, “Điền Chế Độ Đệm”, dùng kết quả do thuật toán “Lần Ngược” tạo ra.

Hình 2 minh họa kết quả soạn hòa âm tự động. Danh sách đầu tiên từ trái sang là các Nốt Quan Trọng trong từng nhịp, mỗi nhịp có tối đa hai nốt. Danh sách thứ hai từ trái sang là Các Nốt Khác trong từng nhịp, mỗi nhịp có tối đa 16 nốt. Danh sách thứ ba từ trái sang là các hợp âm đã tìm được cho từng nhịp, mỗi nhịp dùng một hợp âm, kể cả các nhịp không có giai điệu.

Các thuật toán sẽ dễ trình bày, và cũng sẽ dễ hiểu hơn khi nó được cài đặt bằng một ngôn ngữ lập trình nào đó. Tác giả minh họa các thuật toán trên, viết bằng ngôn ngữ C++, dùng phông chữ Courier, để dễ xem. Đó là các đoạn trích ra từ phần mềm minh họa “Make Chord Algorithms” do tác giả viết, đã được biên dịch và chạy tốt. Phần mềm minh họa “Make Chord Algorithms V2.22” thực hiện các giải thuật vừa nói trên ở mức cơ bản, đơn giản, dễ hiểu cho các giai điệu đầu vào có Key là “C Major” (Đô Trưởng) hoặc “A Minor” (La Thứ). Các phiên bản sau phức tạp hơn, chạy được với mọi Key, với các nhịp thường dùng: 2/4, 3/4, 4/4, 6/8.

Giải thuật Tự Động Soạn Hòa Âm
(Tìm một hợp âm cho mỗi nhịp trong toàn bài)

Ngôn ngữ lập trình: C++, 32 bit, Microsoft Visual Studio 2010.

```
//Các cấu trúc và bảng hoặc biến sẽ sử dụng
int DaUuTien[7];
int Hs1 = 7;//Các hệ số chọn theo kinh nghiệm
int Hs2 = 2;//Các hệ số chọn theo kinh nghiệm
int Hs3 = 2;//Các hệ số chọn theo kinh nghiệm
int Hs4 = 2;//Các hệ số chọn theo kinh nghiệm
int LoadMdOk = 0;
int ModeSimp = 0;
int ModeNorm = 1;
int ModeComp = 0;
int g_RadioVal1 = 1;//= 1 Khi chọn Vr1, = 0 khi chọn Vr2
int g_RadioVal2 = 0;//= 0 Khi chọn Vr1, = 1 khi chọn Vr2
//-----
//Cấu trúc chứa thông tin thời gian để tính chiều dài của nốt
struct NoteLens {
    long OnTime;
    long OfTime;
    long Length;
    int Ev_Idx;
};
NoteLens BngLens[12][2];
//-----
//Bảng dùng để tính điểm chuyển hợp âm
//Các ô có giá trị bằng 1 sẽ được gán khi chọn
complex/normal/simple
BYTE ChyHa_IJ[7][7]={
//j = 0,1,2,3,4,5,6,
    {0,0,0,0,0,0,0},//i=0
    {0,0,0,0,0,0,0},//i=1
    {0,0,0,0,0,0,0},//i=2
    {0,0,0,0,0,0,0},//i=3
    {0,0,0,0,0,0,0},//i=4
    {0,0,0,0,0,0,0},//i=5
    {0,0,0,0,0,0,0},//i=6
};
//-----
//Cấu trúc chứa thông tin về hợp âm
struct StHopAm{
    char HaTxt[7]; //Ký hiệu hợp âm, ví dụ "Am"
    short Note1; //Cao độ nốt 1
    short Note3; //Cao độ nốt 3
    short Note5; //Cao độ nốt 5
    short Note7; //Cao độ nốt 7
```

```

};
StHopAm Ha11,Ha12,Ha13,Ha14,Ha15,Ha16,Ha17;
StHopAm BangHa1[7]; //Bảng chứa 7 hợp âm cho thang âm đang
dùng
//
short NoteQTr[255][ 2]; //Bảng chứa 2 Nốt Quan Trọng, cho
tối đa 255 nhịp
short NoteMor[255][16]; //Bảng chứa tối đa 16 Các Nốt
Khác, cho tối đa 255 nhịp
short Ha_IdxS[255][ 2];
short Ha_Nhip[255][ 4]; //Bảng chứa 4 cao độ các nốt hợp
âm, cho tối đa 255 nhịp
int Node[255][7][ 3]; //Bảng chứa các thông tin của giải
thuật LănNguợc
//
int IsMajor;
int LinCnt1;
int LinCnt2;
int LapCnt = 0;
int g_VarCnt = 1;
int ChyHa_v1Cnt;
//

```

Thuật toán tìm hai "Nốt Quan Trọng" và "Các Nốt Khác" trong từng nhịp

```

//Thuật toán này được gọi sau khi đã nạp thành công
Input.mid (giai điệu đầu vào)
//Các thông tin On/Off của các nốt giai điệu đã được đưa vào
danh sách m_ListEvn, //chính xác theo thứ tự thời gian.
//Kết quả dạng text được đưa vào danh sách m_ListQTr và
m_ListMor
//Kết quả dạng số được đưa vào bảng NoteQTr[][] và
NoteMor[][]
//Thuật toán này tìm được Nốt Quan Trọng và Các Nốt Khác cả
trong trường hợp nốt được //On ở nhịp này, nhưng Off ở nhịp
khác.
//Key của giai điệu đầu vào được xác định bằng nốt cuối
cùng. Bắt buộc là Đô (60) cho //Key Đô Trưởng, là La (69)
cho Key La Thứ. Không hỗ trợ các Key khác.
//Nhịp 4/4, không hỗ trợ các nhịp khác.
void LayNotQTrAndMor(){
    int l;
    long lTimT;
    int IsOfT;
    int NotVT;
    int PrCnt;
    int PrMor;
    int NoEvs;
    int i,j,k;
    int MsCnt;
    int MsNxt;

```

```

int iMor;
int kq;
long CurlTime;
long DeltTime;
long LgTimeMx;
long MsBgTime;
long MsMdTime;
long MsEdTime;
int IsOff;
int NotVl;
int NoQt1;
int Note_QT1;
int Note_QT2;
int NoteLast = 0;

short BngVMor[12];
short BngOnOf[12];

CString Str;
char TempStr[ 80];
char TempSt_[ 80];
char LisQStr[ 80];
char ListStr[160];

if(LoadMdOk == 0) return;

m_ListQTr.ResetContent();
m_ListMor.ResetContent();

MsCnt = 0;
PrCnt = 0;
PrMor = 0;
NoEvs = 0;

for(j=0; j<12; j++){
    BngOnOf[j] = 0;
    BngVMor[j] = 0;
    BngLens[j][0].Length = 0; BngLens[j][0].OnTime = -
1;
    BngLens[j][0].OfTime = -1; BngLens[j][0].Ev_Idx = -1;
    BngLens[j][1].Length = 0; BngLens[j][1].OnTime = -
1;
    BngLens[j][1].OfTime = -1; BngLens[j][1].Ev_Idx = -1;
}
for(j=0; j<16; j++){
    NoteMor[MsCnt][j] = 0;
}

iMor = 0;
Str.Format(_T("%03d:"), MsCnt+1);
strcpy_s(ListStr, Str);

```

```

strcpy_s(LisQStr, Str);

MsBgTime = (long)(MesuLen * MsCnt);
MsMdTime = MsBgTime + MesuLen/2;
MsEdTime = MsBgTime + MesuLen;

i = 0;
while(1){
    MsNxt = 0;
    while(i < EvCount){
        m_ListEvn.GetText(i, TempStr);
        NotVl = 0;
        kq = sscanf_s(TempStr, _T("%06ld:%d:%03d;"),
&CurlTime, &IsOff, &NotVl);

        if(kq == 3){
            if((MsBgTime <= CurlTime) && (CurlTime <
MsEdTime)){
                if(!IsOff){//Nốt On
                    if(NotVl > 0){
                        while(NotVl < 60){
                            NotVl += 12;
                        }
                        while(NotVl > 71){
                            NotVl -= 12;
                        }
                        k = NotVl - 60;

                        //Nốt Mor On
                        ++BngOnOf[k];
                        if(iMor < 16){
                            BngVMor[k] = 1;
                            NoteMor[MsCnt][iMor] =

NotVl;

                            Str.Format(_T("%02d;"),

NotVl);

                            strcat_s(ListStr, Str);
                            ++iMor;
                        }

                        //Nốt QTr On
                        if(CurlTime < MsMdTime){
                            BngLens[k][0].OnTime =

CurlTime;

                            BngLens[k][0].Ev_Idx =

i;

                        }
                        else{
                            BngLens[k][1].OnTime =

CurlTime;

```

```

                                BngLens[k][1].Ev_Idx =
i;
                                }
                                }
                                }
                                else{//Nốt Off
                                    if(NotVl > 0){
                                        while(NotVl < 60){
                                            NotVl += 12;
                                        }
                                        while(NotVl > 71){
                                            NotVl -= 12;
                                        }
                                        k = NotVl - 60;

                                        //Nốt QTr Off
                                        if(CurlTime < MsMdTime){
                                            BngLens[k][0].OfTime =
CurlTime;
                                            BngLens[k][0].Ev_Idx =
i;
                                            if(BngLens[k][0].OnTime
> -1){
                                                DeltTime =
BngLens[k][0].OfTime - BngLens[k][0].OnTime;

                                                BngLens[k][0].Length += DeltTime;

                                                BngLens[k][0].OnTime = -1;

                                                BngLens[k][0].OfTime = -1;

                                                BngLens[k][0].Ev_Idx = -1;
                                            }
                                        }
                                    }
                                }
                                else{
                                    BngLens[k][1].OfTime =
CurlTime;
                                    BngLens[k][1].Ev_Idx =
i;
                                    if(BngLens[k][1].OnTime
> -1){
                                        DeltTime =
BngLens[k][1].OfTime - BngLens[k][1].OnTime;

                                        BngLens[k][1].Length += DeltTime;

                                        BngLens[k][1].OnTime = -1;

                                        BngLens[k][1].OfTime = -1;

```

```

BngLens[k][1].Ev_Idx = -1;
    }
}

//Nốt Mor Off
--BngOnOf[k];
if(iMor < 16){

    if( BngVMor[k] == 0){
        BngVMor[k] = 1;
        ++PrMor;

NoteMor[MsCnt][iMor] = NotVl;

Str.Format(_T("%02d;"), NotVl);

strcat_s(ListStr,Str);

        ++iMor;
    }
}
}
}
else{
    m_SysMess.SetWindowTextA("Out of
MsXxTime");

    MsNxt = 1;
    break;
}
}
else{
    m_SysMess.SetWindowTextA("kq != 3");
    break;
}

++i;
}

if(strlen(ListStr) < 5) strcat_s(ListStr,"00;");
m_ListMor.AddString(ListStr);

for(j=0; j<12; j++){
    //Test 1: No On Time QTr1
    //Set pre-mesur BngLens[j][0].OnTime
    if(BngLens[j][0].OnTime == -1){
        if(BngLens[j][0].OfTime > -1){
            //Tìm NoteOn có cao độ (60 + j) gần
nhất, trước đó

            l = BngLens[j][0].Ev_Idx - 1;

```

```

        while(l >= 0){
            m_ListEvn.GetText(l, TempSt_);
            kq = sscanf_s(TempSt_,
_T("%06ld:%d:%03d;"), &lTimT, &IsOfT, &NotVT);
            if(kq != 3){

                m_SysMess.SetWindowTextA("Error 11: Finding pre
NoteOn");

                return;
            }
            if(!IsOfT){
                while(NotVT > 71) NotVT -=
12;

                while(NotVT < 60) NotVT +=
12;

                if(NotVT == (60 + j)){
                    //Gán cho
                    BngLens[j][0].OnTime =
lTimT;

                    ++PrCnt;
                    break;
                }
            }

            --l;
        }
        if(l < 0){
            m_SysMess.SetWindowTextA("Error
12: Finding pre NoteOn");
            return;
        }
    }

    //Test 2: No On Time QTr2
    //Set pre-mesur BngLens[j][1].OnTime
    if(BngLens[j][1].OnTime == -1){
        if(BngLens[j][1].OfTime > -1){
            //Tìm NoteOn có cao độ (60 + j) gần
nhất, trước đó

            l = BngLens[j][1].Ev_Idx - 1;
            while(l >= 0){
                m_ListEvn.GetText(l, TempSt_);
                kq = sscanf_s(TempSt_,
_T("%06ld:%d:%03d;"), &lTimT, &IsOfT, &NotVT);
                if(kq != 3){

                    m_SysMess.SetWindowTextA("Error 21: Finding pre
NoteOn");

                    return;
                }
            }
        }
    }

```

```

    }
    if(!IsOfT){
        while(NotVT > 71) NotVT -=
12;
        while(NotVT < 60) NotVT +=
12;

        if(NotVT == (60 + j)){
            //Gán cho
BngLens[j][1].OnTime
            BngLens[j][1].OnTime =
lTimT;
            ++PrCnt;
            break;
        }
    }

    --l;
}
if(l < 0){
    m_SysMess.SetWindowTextA("Error
22: Finding pre NoteOn");
    return;
}
}

    }

    NoQt1 = 0;
    //1.No Off Time
    if(BngLens[j][0].OfTime == -1){
        if(BngLens[j][0].OnTime > -1)
            BngLens[j][0].Length += (MsMdTime -
BngLens[j][0].OnTime);
    }
    if(BngLens[j][1].OfTime == -1){
        if(BngLens[j][1].OnTime > -1)
            BngLens[j][1].Length += (MsEdTime -
BngLens[j][1].OnTime);
    }

    //2.No On Time QTr1
    if(BngLens[j][0].OnTime == -1){
        //2.1.Nhung có Off time
        if(BngLens[j][0].OfTime > -1){
            //Coi On Time là MsBgTime
            BngLens[j][0].OnTime = MsBgTime;
            BngLens[j][0].Length +=
(BngLens[j][0].OfTime - MsBgTime);
        }
        //2.2.No Off Time
        else{

```

```

        NoQt1 = 1;
    }
}

//2.No On Time QTr2
if(BngLens[j][1].OnTime == -1){
    //2.1.Nhung có Off time
    if(BngLens[j][1].OfTime > -1){
        //2.1.1.Nếu chưa có QTr1
        if(NoQt1){
            //Thì gán trị này cho QTr1
            BngLens[j][0].Length +=
(BngLens[j][1].OfTime - MsBgTime);
        }
        //2.1.2.Nếu đã có QTr1
        else{
            //Thì gán trị này cho QTr2
            BngLens[j][1].Length +=
(BngLens[j][1].OfTime - MsMdTime);
        }
    }
}

}

LgTimeMx = 0; Note_QT1 = 0;
for(j=0; j<12; j++){
    if(BngLens[j][0].Length > LgTimeMx){
        LgTimeMx = BngLens[j][0].Length;
        Note_QT1 = 60 + j;
    }
}
//Loại bỏ các nốt quá ngắn
if(LgTimeMx < 32) Note_QT1 = 0; //96,48

LgTimeMx = 0; Note_QT2 = 0;
for(j=0; j<12; j++){
    if(BngLens[j][1].Length > LgTimeMx){
        LgTimeMx = BngLens[j][1].Length;
        Note_QT2 = 60 + j;
    }
}
//Loại bỏ các nốt quá ngắn
if(LgTimeMx < 32) Note_QT2 = 0; //96,48
//-----

if((Note_QT1 == 0) && (Note_QT2 == 0)){
    //Có ít nhất 1 NoteMor
    if(NoteMor[MsCnt][0] > 0){
        Note_QT2 = NoteMor[MsCnt][0];
    }
    //Không NoteMor nào
}

```

```

else{
    //Tìm trong BngOnOf[12] một giá trị > 0
    (đã On mà chưa Off)
    k = 0;
    while(k<12){
        if(BngOnOf[k] > 0){
            Note_QT1 = 60 + k;
            ++NoEvs;
            break;
        }
        ++k;
    }
}

Str.Format(_T("%02d;%02d;"), Note_QT1, Note_QT2);
if (Note_QT2 > 0) NoteLast = Note_QT2;
else if (Note_QT1 > 0) NoteLast = Note_QT1;

NoteQTr[MsCnt][0] = Note_QT1;
NoteQTr[MsCnt][1] = Note_QT2;
strcat_s(LisQStr, Str);
m_ListQTr.AddString(LisQStr);

if(i >= EvCount){
    Str.Format(_T("End Of Events <PrCnt=%d,
PrMor=%d, NoEvs=%d>"), PrCnt, PrMor, NoEvs);
    m_SysMess.SetWindowTextA(Str);
    break;
}

if(MsNxt){
    MsCnt = MsCnt +1;
    iMor = 0;
    Str.Format(_T("%03d:"), MsCnt+1);
    strcpy_s(ListStr, Str);
    strcpy_s(LisQStr, Str);

    MsBgTime = (long)(MesuLen * MsCnt);
    MsMdTime = MsBgTime + MesuLen/2;
    MsEdTime = MsBgTime + MesuLen;

    for(j=0; j<12; j++){
        BngVMor[j] = 0;
        BngLens[j][0].Length = 0;
        BngLens[j][0].OnTime = -1;
        BngLens[j][0].OfTime = -1; BngLens[j][0].Ev_Idx
        = -1;
        BngLens[j][1].Length = 0;
        BngLens[j][1].OnTime = -1;

```

```

        BngLens[j][1].OfTime = -1; BngLens[j][1].Ev_Idx
        = -1;
    }
    for(j=0; j<16; j++){
        NoteMor[MsCnt][j] = 0;
    }
}
else{
    m_SysMess.SetWindowTextA("No MsNxt");
    break;
}
}

LinCnt1 = m_ListQTr.GetCount();

if(NoteLast == 60){
    IsMajor = 1;
    m_KeyName.SetWindowTextA("C Major");
    m_MakeAcc.EnableWindow(1);
}
else if(NoteLast == 69){
    IsMajor = 0;
    m_KeyName.SetWindowTextA("A Minor");
    m_MakeAcc.EnableWindow(1);
}
else{
    m_KeyName.SetWindowTextA("?");
    m_MakeAcc.EnableWindow(0);
}
}
}

```

Hàm tạo 7 hợp âm của thang C Major

```

void TaoHopAm_kCmajor(){
    //1/C : 60, 64, 67
    strcpy_s(Ha11.HaTxt, "C");
    Ha11.Note1 = 60;
    Ha11.Note3 = 64;
    Ha11.Note5 = 67;
    Ha11.Note7 = 0;

    //2/Dm : 62, 65, 69
    strcpy_s(Ha12.HaTxt, "Dm");
    Ha12.Note1 = 62;
    Ha12.Note3 = 65;
    Ha12.Note5 = 69;
    Ha12.Note7 = 0;

    //3/Em : 64, 67, 71
    strcpy_s(Ha13.HaTxt, "Em");
    Ha13.Note1 = 64;
    Ha13.Note3 = 67;
}

```

```

Ha13.Note5 = 71;
Ha13.Note7 = 0;

//4/F :65,69,60
strcpy_s(Ha14.HaTxt, "F");
Ha14.Note1 = 65;
Ha14.Note3 = 69;
Ha14.Note5 = 60;
Ha14.Note7 = 0;

//5/G7 :67,71,62,65
strcpy_s(Ha15.HaTxt, "G"); //"G7"
Ha15.Note1 = 67;
Ha15.Note3 = 71;
Ha15.Note5 = 62;
Ha15.Note7 = 0; //65;

//6/Am :69,60,64
strcpy_s(Ha16.HaTxt, "Am");
Ha16.Note1 = 69;
Ha16.Note3 = 60;
Ha16.Note5 = 64;
Ha16.Note7 = 0;

//7/Bo :71,62,65
strcpy_s(Ha17.HaTxt, "Bo");
Ha17.Note1 = 71;
Ha17.Note3 = 62;
Ha17.Note5 = 65;
Ha17.Note7 = 0;
}

```

Hàm tạo 7 hợp âm của thang A Minor

```

void TaoHopAm_kAminor(){
    //1/Am :69,60,64
    strcpy_s(Ha11.HaTxt, "Am");
    Ha11.Note1 = 69;
    Ha11.Note3 = 60;
    Ha11.Note5 = 64;
    Ha11.Note7 = 0;

    //2/Bo :71,62,65
    strcpy_s(Ha12.HaTxt, "Bo");
    Ha12.Note1 = 71;
    Ha12.Note3 = 62;
    Ha12.Note5 = 65;
    Ha12.Note7 = 0;

    //3/C :60,64,67
    strcpy_s(Ha13.HaTxt, "C");
    Ha13.Note1 = 60;

```

```

Ha13.Note3 = 64;
Ha13.Note5 = 67;
Ha13.Note7 = 0;

//4/Dm :62,65,69
strcpy_s(Ha14.HaTxt,"Dm");
Ha14.Note1 = 62;
Ha14.Note3 = 65;
Ha14.Note5 = 69;
Ha14.Note7 = 0;

//5/Em7:64,67,71,62
strcpy_s(Ha15.HaTxt,"Em");//"Em7"
Ha15.Note1 = 64;
Ha15.Note3 = 67;
Ha15.Note5 = 71;
Ha15.Note7 = 0;//62;

//6/F :65,69,60
strcpy_s(Ha16.HaTxt,"F");
Ha16.Note1 = 65;
Ha16.Note3 = 69;
Ha16.Note5 = 60;
Ha16.Note7 = 0;

//7/G :67,71,62
strcpy_s(Ha17.HaTxt,"G");
Ha17.Note1 = 67;
Ha17.Note3 = 71;
Ha17.Note5 = 62;
Ha17.Note7 = 0;
}

```

Giải thuật Lăn Ngược

//Giải thuật này lăn ngược từ nhịp cuối đến nhịp đầu tiên của bài; Tính điểm cho mỗi lần chuyển hợp âm từ nhịp này sang nhịp tiếp theo; Ghi nhớ các chuyển tiếp có điểm cao nhất (tối ưu).

```

int LanNguoc(){
    int CurLn;
    int DiemMx;
    int i,j;
    int Kq;
    int Er;
    int D3;
    int T3;

    //Reset bảng DaUuTien[7]
    for(i=0; i<7; i++){
        DaUuTien[i] = 0;
    }
}

```

```

//Đặt hợp âm bậc I tại nhịp cuối
CurLn = LinCntl -1;
Node[CurLn][0][0] = 1000;
Node[CurLn][0][1] = 0;
Node[CurLn][0][2] = 2;
for(i=1; i<7; i++){
    Node[CurLn][i][0] = 0;
    Node[CurLn][i][1] = 0;
    Node[CurLn][i][2] = 0;
}
--CurLn;

//Lần ngược từ cuối về đầu
Er = 0;
while(CurLn >= 0){
    for(i=0; i<7; i++){
        DiemMx = -32000;

        for(j=0; j<7; j++){
            Kq = TinhDiemToiUu(i,j, CurLn, &D3);
            T3 = D3;
            if(Kq == -1){
                Er = 1;
                break;
            }
            //Tăng ưu tiên lặp lại 2 lần hợp âm cùng
            bậc, nếu có thể,
            if(i == j){
                if(NoteQTr[CurLn][0] ==
NoteQTr[CurLn][1]){
                    if(NoteQTr[CurLn][0] ==
NoteQTr[CurLn+1][0]){
                        if(!DaUuTien[i]){
                            DaUuTien[i] = 1;
                            Kq += 30;
                        }
                        else{
                            DaUuTien[i] = 0;
                        }
                    }
                }
                else{
                    DaUuTien[i] = 0;
                }
            }

            Kq += Node[CurLn+1][j][0];
            T3 += Node[CurLn+1][j][2];
            if(g_RadioVal1 == 1){//Khi chọn Vr1
                if(Kq > DiemMx){

```

```

        DiemMx = Kq;
        Node[CurLn][i][0] = DiemMx;
        Node[CurLn][i][1] = j;
        Node[CurLn][i][2] = T3;
    }
}
else{//Khi chọn Vr2
    if(Kq >= DiemMx){
        DiemMx = Kq;
        Node[CurLn][i][0] = DiemMx;
        Node[CurLn][i][1] = j;
        Node[CurLn][i][2] = T3;
    }
}
}
if(Er)    break;
}
if(Er)    break;

--CurLn;
}

if(Er)    return CurLn;
return -1;
}

```

Giải thuật "Điền Hộp Âm" và "Chế Độ Đệm"

```

//Giải thuật này sử dụng kết quả đã được tạo ra bởi giải
thuật LănNguợc;
//Dùng thêm các hộp âm kiểu CSus4, CDim5, CAug5, CAdd7.
//Kết quả dạng text đưa vào danh sách m_ListAcc để hiển thị,
dạng số đưa vào bảng //Ha_Nhip[][] để tạo "Nốt Xác Định Hộp
Âm" trong tập tin điều khiển Output.mid
void DienHopAm(){
    short    HaNs;
    short    HaN5;
    short    HaD5;
    short    HaA5;
    short    HaN7;
    short    MlVl;

    int    MCnt;
    int    HaIx;
    int    CoN7;
    int    CoS4;
    int    CoD5;
    int    CoA5;
    int    i,j;
    int    DiemCr;
    int    DiemMx;
    int    iOk;

```

```

int  jNx;
int  FDr;
int  LCurr;
int  DaF2;
int  DLF2;
int  LnCnt = 0;
int  V1Cnt;
int  TmMx;
int  DmCr;
int  Tmp3[7];
int  iOk1[7];

char Kt_2 = ' ';
char StrBf[10];

CString  Str;
CString  HaStr;

for(i=0; i<7; i++){
    Tmp3[i] = 0;
    iOk1[i] = 0;
}

DiemMx = -32000;
j = 0;
for(i=0; i<7; i++){
    DiemCr = Node[0][i][0];
    if(DiemCr >= DiemMx){
        if(DiemCr > DiemMx){
            DiemMx = DiemCr;
            j = 0;
            iOk1[j] = i;
            Tmp3[j] = Node[0][i][2];
            ++j;
        }
        else{
            iOk1[j] = i;
            Tmp3[j] = Node[0][i][2];
            ++j;
        }
    }
}
g_VarCnt = j;

TmMx = -32000;
for(i=0; i<g_VarCnt; i++){
    DmCr = Tmp3[i];
    if(DmCr >= TmMx){
        TmMx = DmCr;
        iOk = i;
    }
}

```

```

}

VlCnt = DemVlChyHa_IJ();
Str.Format(_T("iOk=%d, DiemMx=%d, TmMx=%d, g_VarCnt=%d,
VlCnt=%d"), iOk, DiemMx, TmMx, g_VarCnt, VlCnt);
m_SysMess.SetWindowTextA(Str);

DaF2 = 0;
DLF2 = 0;
for(LCurr=0; LCurr<LinCnt1; LCurr++){
    FDr = 0;
    HaStr = BangHal[iOk].HaTxt;
    jNx = Node[LCurr][iOk][1];

    if(DLF2 > 8){
        if(iOk == 4 && jNx == 0 ){
            FDr = 2;
            DaF2 = 1;
            DLF2 = 0;
        }
    }

    if(iOk == 0 && DaF2){
        FDr = 1;
        DaF2 = 0;
    }

    if(LCurr == LinCnt1-1) FDr = 3;
    Ha_IdxS[LCurr][0] = iOk;
    Ha_IdxS[LCurr][1] = FDr;

    ++LnCnt;
    ++DLF2;
    iOk = jNx;
}

m_ListAcc.ResetContent();

//Hiệu chỉnh hợp âm, rồi đưa vào danh sách kết quả
for(LCurr=0; LCurr<LinCnt1; LCurr++){
    HaIx = Ha_IdxS[LCurr][0];
    strcpy_s(StrBf, BangHal[HaIx].HaTxt);
    Kt_2 = StrBf[1];

    //1.Nốt 7
    HaN7 = BangHal[HaIx].Notel - 2;
    if(HaN7 < 60) HaN7 += 12;
    if(HaN7 > 71) HaN7 -= 12;

    MCnt = 0;
    CoN7 = 0; i = 0;

```

```

while(i<16){
    MlVl = NoteMor[LCurr][i];
    if(MlVl > 0) ++MCnt;
    if(MlVl == HaN7){
        CoN7 = 1;
        break;
    }
    ++i;
}
//Gán nốt 7 cho hợp âm bậc 5 nếu không có nốt giai
điệu
if((HaIx == 4) && (MCnt == 0)) CoN7 = 1;

//2.Sus4
CoS4 = 0;
//Không làm khi hợp âm là Dim
if((Kt_2 != 'o')){

HaNs = BangHal[HaIx].Notel + 5;
if(HaN7 > 71) HaNs -= 12;
i = 0;
while(i<16){
    MlVl = NoteMor[LCurr][i];
    if(MlVl == HaNs){
        CoS4 = 1;
        break;
    }
    ++i;
}
}

//3.Dim5/Aug5
CoD5 = 0;
CoA5 = 0;
if((Kt_2 != 'm') && (Kt_2 != 'o' && (HaIx != 0))){
    HaD5 = BangHal[HaIx].Notel + 6;
    if(HaD5 > 71) HaD5 -= 12;

    HaA5 = BangHal[HaIx].Notel + 8;
    if(HaA5 > 71) HaA5 -= 12;

    i = 0;
    while(i<16){
        MlVl = NoteMor[LCurr][i];
        if(MlVl == HaD5){
            CoD5 = 1;
            break;
        }
        ++i;
    }
}

```

```

        i = 0;
        while(i<16){
            MlVl = NoteMor[LCurr][i];
            if(MlVl == HaA5){
                CoA5 = 1;
                break;
            }
            ++i;
        }
    }

    //-Loại bỏ trường hợp CoS4 và CoD5 mà giá trị HaNs
trùng HaD5
    //Ví dụ với hợp âm "Bo", khi CoS4 và CoD5 sẽ là
    "Bs-" => HaNs trùng HaD5
    //-Không làm khi ModeSimp == 1
    if(ModeSimp || (HaNs == HaD5)){
        CoS4 = CoD5 = 0;
    }
    //Tạo hợp âm đã chỉnh lại
    if(CoS4){
        StrBf[1] = 's';
        StrBf[2] = 0;
    }
    else{
        HaNs = BangHal[HaIx].Note3;
    }

    if(CoD5){
        strcat_s(StrBf, "-");
        HaN5 = HaD5;
    }
    else if(CoA5){
        strcat_s(StrBf, "+");
        HaN5 = HaA5;
    }
    else{
        HaN5 = BangHal[HaIx].Note5;
    }

    if(CoN7) strcat_s(StrBf, "7");
    else      HaN7 = 0;

    //Luu vào Ha_Nhip[][]
    if(LCurr == LinCntl -1){//Không sửa tại nhịp cuối
        strcpy_s(StrBf, BangHal[HaIx].HaTxt);
        Ha_Nhip[LCurr][0] = BangHal[HaIx].Note1;
        Ha_Nhip[LCurr][1] = BangHal[HaIx].Note3;
        Ha_Nhip[LCurr][2] = BangHal[HaIx].Note5;
        Ha_Nhip[LCurr][3] = BangHal[HaIx].Note7;
    }

```

```

    }
    else{
        Ha_Nhip[LCurr][0] = BangHa1[HaIx].Note1;
        Ha_Nhip[LCurr][1] = HaNs;
        Ha_Nhip[LCurr][2] = HaN5;
        Ha_Nhip[LCurr][3] = HaN7;
    }

    FDr    = Ha_IdxS[LCurr][1];
    Str.Format(_T("%03d: %d  %s"), LCurr+1, FDr,
StrBf);
    m_ListAcc.AddString(Str);
}

LinCnt2 = LnCnt;
if(LinCnt2 != LinCnt1){
    Str.Format(_T("Error: LinCnt1=%d != LinCnt2=%d"),
LinCnt1, LinCnt2);
    m_SysMess.SetWindowTextA(Str);
    m_CreateMidi.EnableWindow(0);
}
else{
    m_CreateMidi.EnableWindow(1);
}
}

```

Hàm "Đánh Giá Độ Tối Ưu"" dùng trong giải thuật Lăn Ngược

```

int TinhDiemToiUu(int ic, int jn, int LCurr, int* d3){
    int    i;
    int    Kq1;
    int    Kq2;
    int    Kq3;
    int    Kq4;
    int    dvd;
    int    CoM;

    int    T3 = 0;

    short    MlV1,MlV2;
    short    Vc7 , Vn7;
    short    mVl , cNv;

    //Lấy 2 NoteQTr tại nhịp LCurr để kiểm tra HA (Hợp Âm)
    MlV1 = NoteQTr[LCurr][0];
    MlV2 = NoteQTr[LCurr][1];

    Kq3 = 0;
    if((MlV1 != 0) || (MlV2 != 0)){
        //Điểm 3: điểm trùng nốt của hợp âm với NoteQTr
        if(MlV1 > 0){
            if(MlV1 == BangHa1[ic].Note1)    ++Kq3;

```

```

        if(MlV1 == BangHa1[ic].Note3)    ++Kq3;
        if(MlV1 == BangHa1[ic].Note5)    ++Kq3;
        if(MlV1 == BangHa1[ic].Note7)    ++Kq3;
    }
    if(MlV2 > 0){
        if(MlV2 == BangHa1[ic].Note1)    ++Kq3;
        if(MlV2 == BangHa1[ic].Note3)    ++Kq3;
        if(MlV2 == BangHa1[ic].Note5)    ++Kq3;
        if(MlV2 == BangHa1[ic].Note7)    ++Kq3;
    }
    if(Kq3 == 0){
        *d3 = 0;
        return 0;
    }

    T3 = Kq3;
}
else{
    Kq3 = 1;
}

//Điểm 4: điểm trùng nốt của hợp âm với các nốt giai
điệu khác
Kq4 = 0;
i = 0;
CoM = 0;
while(i<16){
    mVl = NoteMor[LCurr][i];
    if(mVl > 0){
        CoM = 1;
        if(mVl == BangHa1[ic].Note1)    ++Kq4;
        if(mVl == BangHa1[ic].Note3)    ++Kq4;
        if(mVl == BangHa1[ic].Note5)    ++Kq4;
        if(mVl == BangHa1[ic].Note7)    ++Kq4;
    }
    ++i;
}
if(CoM && (Kq4 == 0))    return 0;

if(T3 > 0)    T3 += Kq4;

//Điểm 1: điểm chuyển hợp âm theo "kinh nghiệm chuyên
gia"
if((ic == 4) && (jn == 0))    Kq1 = 1;
else    Kq1 = 0;
if((MlV1 == 0) && (MlV2 == 0))    dvd = 2;
else    dvd = 1;
if(ChyHa_IJ[ic][jn] > 0)    Kq1 += dvd;

//Điểm 2: điểm trùng nốt của hai hợp âm
Kq2 = 0;

```

```

Vn7 = BangHal[jn].Note7;
Vc7 = BangHal[ic].Note7;

cNv = BangHal[ic].Note1;
if      (cNv == BangHal[jn].Note1)      Kq2 += 1;
else if (cNv == BangHal[jn].Note3)      Kq2 += 1;
else if (cNv == BangHal[jn].Note5)      Kq2 += 1;
else if (Vn7 != 0){
    if(cNv == Vn7)                      Kq2 += 1;
}

cNv = BangHal[ic].Note3;
if      (cNv == BangHal[jn].Note1)      Kq2 += 1;
else if (cNv == BangHal[jn].Note3)      Kq2 += 1;
else if (cNv == BangHal[jn].Note5)      Kq2 += 1;
else if (Vn7 != 0){
    if(cNv == Vn7)                      Kq2 += 1;
}

cNv = BangHal[ic].Note5;
if      (cNv == BangHal[jn].Note1)      Kq2 += 1;
else if (cNv == BangHal[jn].Note3)      Kq2 += 1;
else if (cNv == BangHal[jn].Note5)      Kq2 += 1;
else if (Vn7 != 0){
    if(cNv == Vn7)                      Kq2 += 1;
}

if(Vc7 != 0){
if      (Vc7 == BangHal[jn].Note1)      Kq2 += 1;
else if (Vc7 == BangHal[jn].Note3)      Kq2 += 1;
else if (Vc7 == BangHal[jn].Note5)      Kq2 += 1;
else if (Vc7 == BangHal[jn].Note7)      Kq2 += 1;
}

*d3 = T3;

return (Hs1*Kq1 + Hs2*Kq2 + Hs3*Kq3 + Hs4*Kq4);
}

```

Hàm tạo các hợp âm kết quả

```

void FillChords(){
    int      Kq;
    CString  Str;

    if(LinCnt1 == 0)    return;

    if(LinCnt1 > 250){
        m_SysMess.SetWindowTextA("The input melody has too
many measures");
        return;
    }
}

```

```

m_MakeAcc.EnableWindow(0);
ChyHa_v1Cnt = ResetChyHa_IJ();

if(IsMajor)    TaoHopAm_kCmajor();
else          TaoHopAm_kAminor();

BangHa1[0] = Ha11;
BangHa1[1] = Ha12;
BangHa1[2] = Ha13;
BangHa1[3] = Ha14;
BangHa1[4] = Ha15;
BangHa1[5] = Ha16;
BangHa1[6] = Ha17;

LapCnt = 0;
Kq      = LanNguoc();
if(Kq != -1){
    Str.Format(_T("LanNguoc() error: line %03d"),
Kq+1);
    m_SysMess.SetWindowTextA(Str);
    m_MakeAcc.EnableWindow(1);
    return;
}

DienHopAm();
}

```

Hàm đặt chế độ "Đơn Giản"(Simple)/"Bình Thường"(Normal)/"Phức Tạp"(Complex)

/*Hàm này gán các ô có giá trị bằng 1 vào bảng ChyHa_IJ[7][7] theo kinh nghiệm chuyên gia. Cách gán có thể khác nhau tùy thể loại nhạc. Ở đây tác giả sử dụng cách gán cho kết quả tốt với nhạc Dân Ca Việt Nam, nhạc dance, pop, rock,... quốc tế. Các biến ModeSimp, ModeNorm, ModeComp được gán trị khi người dùng chọn độ phức tạp soạn hòa âm tự động. Biến IsMajor được tự động đặt bằng 1 nếu giai điệu đầu vào (trong Input.mid) là Đô Trưởng.

*/

```

int ResetChyHa_IJ(){
    int i,j;
    int V1Cnt = 0;

    for(i=0; i<7; i++){
        for(j=0; j<7; j++){
            ChyHa_IJ[i][j] = 0;
        }
    }
}

```

```

if(ModeSimp){
    ChyHa_IJ[0][3] = 1;// I - IV
    ChyHa_IJ[3][4] = 1;// IV - V
    ChyHa_IJ[4][0] = 1;// V - I
    V1Cnt = 3;
}
else if (ModeNorm){
    ChyHa_IJ[0][3] = 1;// I - IV
    ChyHa_IJ[3][4] = 1;// IV - V
    ChyHa_IJ[4][0] = 1;// V - I

    ChyHa_IJ[0][2] = 1;// I - III
    ChyHa_IJ[2][3] = 1;//III - IV

    V1Cnt = 5;
}
else{
    ChyHa_IJ[0][3] = 1;// I - IV
    ChyHa_IJ[3][4] = 1;// IV - V
    ChyHa_IJ[4][0] = 1;// V - I

    ChyHa_IJ[0][2] = 1;// I - III
    ChyHa_IJ[2][3] = 1;//III - IV

    if(IsMajor)    ChyHa_IJ[0][5] = 1;    // I - VI
    ChyHa_IJ[5][1] = 1;    // VI - II
    ChyHa_IJ[5][4] = 1;    // VI - V
    if(IsMajor)    ChyHa_IJ[5][3] = 1;    // VI - IV
    ChyHa_IJ[1][4] = 1;// II - V

    ChyHa_IJ[3][1] = 1;// IV - II

    V1Cnt = 9;
    if(IsMajor)    V1Cnt += 2;
}

return V1Cnt;
}

```

Yêu cầu bảo hộ

1. Phương pháp hòa âm tự động bao gồm các bước:

nhận các nốt giai điệu của người dùng để tạo một tập tin giai điệu tương ứng bằng một phần mềm có sẵn “Encore”, trong đó người dùng sẽ nhập các nốt nhạc chính của giai điệu từ một bản nhạc in trên giấy hoặc do chính người dùng sáng tác ra, xuất bản nhạc đó ra tập tin loại MIDI có đuôi là “.mid”;

tạo tập tin điều khiển thông qua việc phân tích tập tin giai điệu bằng phần mềm “Midi Utility”;

điều khiển một thiết bị MIDI có khả năng đệm tự động sử dụng các thông tin trong tập tin điều khiển bằng phần mềm “MidiOut Player”;

thực hiện các lệnh điều khiển từ phần mềm “MidiOut Player” gửi đến để tạo ra âm nhạc và ghi lại thành tập tin âm nhạc bởi thiết bị MIDI có khả năng đệm tự động;

trong đó, thiết bị MIDI là một loại đàn điện tử có khả năng đệm tự động;

trong đó, “Encore” được tạo cấu hình để thực hiện các công việc: nhận các nốt nhạc và lời hát do người dùng nhập vào, hoặc lấy các nốt nhạc từ một tệp MIDI có sẵn; in ra giấy (văn bản) các nốt nhạc và lời hát đó trên khuôn nhạc; xuất ra tập tin MIDI (Input.mid) chứa các nốt nhạc do người dùng đã nhập vào;

trong đó, “Midi Utility” được tạo cấu hình để thực hiện các công việc: lấy các nốt nhạc của giai điệu trong tệp MIDI (Input.mid) do “Encore” tạo ra; tự động xác định giọng (key) của giai điệu; tự động soạn hòa âm cho tất cả các nhịp; cho người dùng chọn một /vài kiểu đệm trong các kiểu đã soạn trước, kèm theo; tạo ra phần đệm tự động gồm nhiều nhạc cụ tham gia; xuất kết quả đệm tự động ra tệp MIDI (Output.mid); nếu người dùng muốn dùng các kiểu đệm trên đàn Organ, thì “Midi Utility” tạo ra tệp dạng Output.mid chỉ có các nốt giai điệu, và các lệnh điều khiển đàn;

trong đó, “MidiOut Player” được tạo cấu hình để thực hiện các công việc sau: lấy các nốt giai điệu và các lệnh điều khiển đàn do “Midi Utility” tạo ra khi

người dùng muốn dùng các kiểu đệm trên đàn (Output.mid); cho người dùng chọn loại đàn có khả năng đệm tự động; chuyển các lệnh điều khiển đàn của Midi Utility thành các lệnh điều khiển loại đàn đã chọn; sắp xếp theo thứ tự thời gian với độ chính xác đến 1 mili giây (ms); chèn thêm các xung đồng bộ vào giữa các nội dung lấy từ tệp Output.mid đã nói trên; gửi lần lượt (theo thời gian thực) các nội dung đó đến đàn qua cáp MIDI In/Out hoặc cáp USB.

2. Phương pháp hòa âm tự động theo điểm 1, trong đó bước tạo tập tin điều khiển bao gồm hai giải thuật xử lý:

giải thuật phát hiện giọng (còn gọi là điệu thức/khóa) của giai điệu đầu vào; và

giải thuật tự động soạn hòa âm cho mọi ca khúc.

3. Phương pháp hòa âm tự động theo điểm 2, trong đó thông tin trong tập tin điều khiển bao gồm:

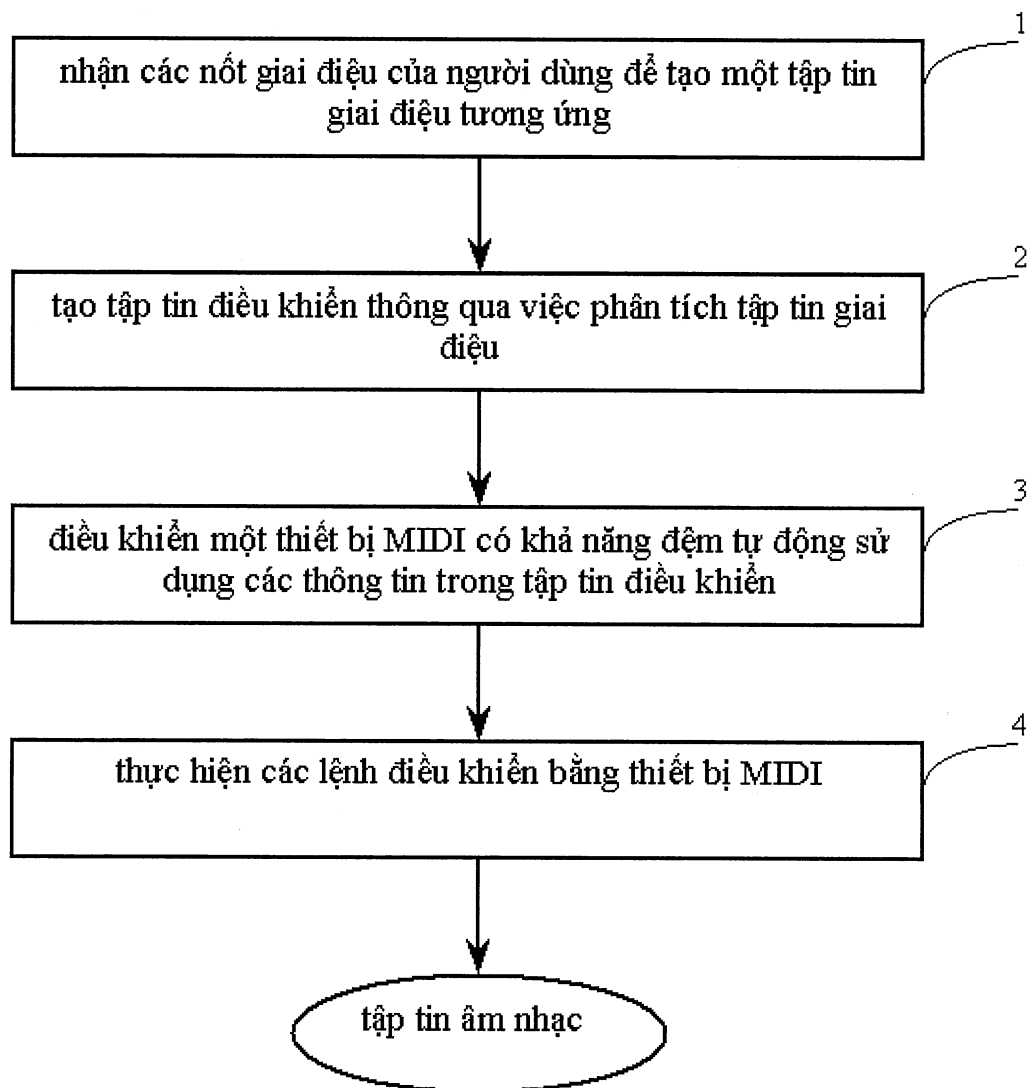
các chế độ đệm trên đàn thường có, là các chế độ: Dẫn Nhập (Intro), Kết Thúc (Ending), Dồn Trống (Fill-In), và “Phần Chính” (Main);

ngoài các thông tin đã nói trên, phần mềm “MidiOut Player” còn gửi đến đàn các lệnh điều khiển khác:

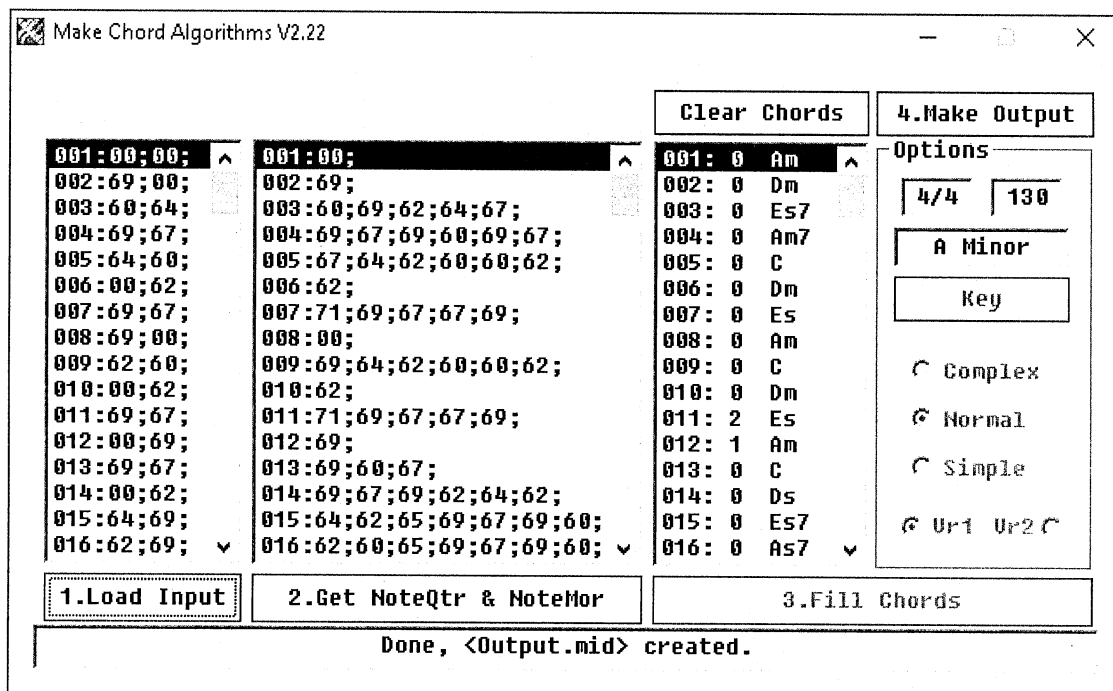
- lệnh chọn nhịp điệu (Select Style);
- lệnh “Bắt Đầu” (Start), để khởi động hệ thống tự động đệm của đàn;
- lệnh “Dừng” (Stop), để đàn dừng (stop) chế độ tự động đệm;
- các xung đồng bộ (Clock), các xung này được gửi đều đặn, liên tục đến đàn với sai số nhỏ hơn 1 milli giây.

4. Phương pháp hòa âm tự động theo điểm 3, trong đó:

khi thực hiện các lệnh điều khiển từ phần mềm “MidiOut Player” gửi đến để tạo ra âm nhạc và ghi lại thành tập tin âm nhạc, thiết bị MIDI có khả năng đệm tự động luôn nhận được các xung đồng bộ là các mốc thời gian, căn cứ vào các mốc thời gian đó, thiết bị MIDI tự động điều chỉnh tốc độ (Tempo) của bộ đệm.



Hình 1



Hình 2

1 - DiemXua

2 - Chords

3 - ChdCmd

4 - Ak-Drums

Hình 3