Jazzification System Design

26/Jun/2022 Tongyu Lu

Basic Elements in a Jazz Trio

In a typical jazz trio (bass, drum, piano) playing swing jazz, bass plays half-time or walking bass, drum plays style-specific groove, and piano plays voicing and melody/solo.

Now, we want to design a rule-based system to transfer any piece in 3/4 or 4/4 meter into trio-styled swing jazz. How to achieve this?
We have to implement a bass generator and a voicing generator, and we have to swingify the melody. When the three parts are done, we are able to swingify a piece.

- Bass design: normally, we play walking bass where notes are chosen from $\{\text{root}, 5^{\text{th}}, \text{leading}\}$ notes. However, this default pattern often changes when chord quality changes (e.g., for diminished chords, we play tritone rather than 5^{th} note). As a result, we need to develop a chord symbol identification system ahead of time before designing the bass generator. In this article, the chord symbol identification system is assumed to be already existing.
- Voicing design: given a chord track, we have to first extend the chords (e.g., triads) into jazz style (normally seventh chords). And then, we define a mapping from chord to its corresponding rootless-voicings. Finally, we have to design the voicing groove rhythm pattern.
- Swingify the melody: it is simple, just alter the onset/duration of 8^{th} notes.

We start from the simplest one: swingify the melody.

Swingify the Melody

Although this procedure seems straightforward, there are several key points to take care. They are summarized into the following algorithm.

Algorithm: swingify melody

Input: a melody track $[n_i = (t_i, \Delta t_i, p_i), i = 1, \dots, N]$, the first downbeat time t_0
Output: swingified melody track $[n_i' = (t_i', \Delta t_i', p_i), i = 1, \dots, N]$
(A fourth note has duration 24.)

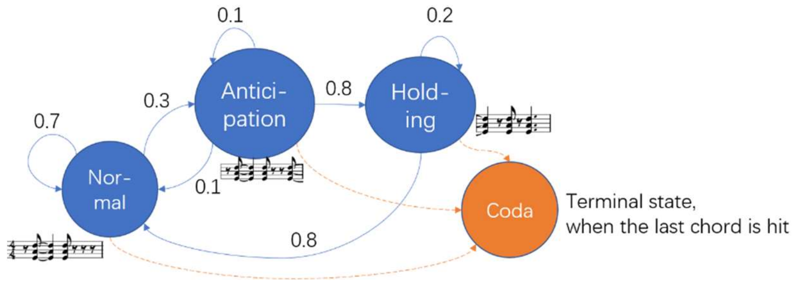
1. Initiate $\text{prolonged\_flag} = \text{False}$
2. For $i = 1, \dots, N$:
3. Beat time $\tilde{t}_i = t_i - t_0$
4. If $\Delta t_i \bmod 24 = 12$ (if n_i represents an 8^{th} note or equivalent):
5. If $\tilde{t}_i \bmod 24 = 0$ and ($\Delta t_{i+1} \bmod 24 = 12$ or $\Delta t_{i+1} \bmod 24 = 0$):
6. $\Delta t_i = \Delta t_i + 4$
7. $\text{prolonged\_flag} = \text{True}$
8. Elif $\tilde{t}_i \bmod 24 = 12$ and prolonged_flag:
9. $\Delta t_i = \Delta t_i - 4$
10. $t_i = t_i + 4$
11. $\text{prolonged\_flag} = \text{False}$
12. Elif $\Delta t_i \bmod 24 = 0$:
13. If $\tilde{t}_i \bmod 24 = 12$ and prolonged_flag:

$t_i = t_i + 4$
prolonged_flag $=$ True
Return the modified melody track

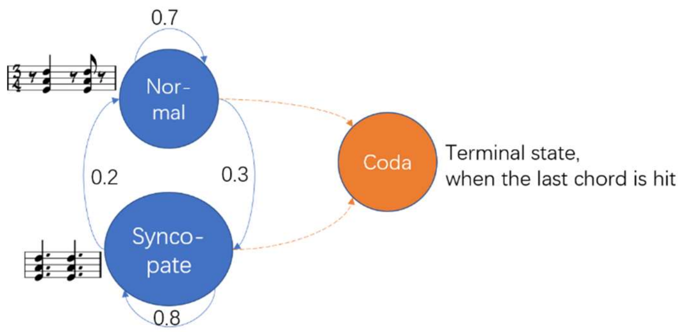
Voicing Design

Before voicing design, one should have already converted the given chord progression into jazzy-extended version. Now, the central problem is to design the rhythm of voicings, and decide “which time to play which voicing”.
We design the rhythm based on a Markov chain. We maintain a dictionary of swing rhythm patterns beforehand, including the “normal rhythm templates”, “anticipation rhythm templates”, “holding rhythm templates” and “coda rhythm templates”. The finite-transition graphs is as follows:

- For 4/4 meter:



- For 3/4 meter:



The scores in the figures above are just illustrating examples. In implementation, we defined multiple rhythm templates for each rhythm class.
Given a piece of M measures, run transitions of the Markov model for M times, obtaining M rhythm templates. Then, translate the M templates into a note onset-duration track. Finally, align chord types to each voicing, and plug in its corresponding voicing to that time slot.
Check the code for details.

Bass Design

We implemented walking bass generator as an example. There is no need to design a Markov model for walking bass rhythm, because its rhythm is almost determined: all 4th notes. The central problem is to choose the bass pitches.
The bass pitch choice is also a Markov model. However, it is more complicated. The set of rules are summarized in the following algorithm:

Algorithm: determine the next bass note

Input: previous bass pitch p_{i-1}, direction (contour) of bassline d_{i-1} (“STAY”, “UP” or “DOWN”), bass note start-time τ_i, chord position i_c, max bass pitch P^M, min bass pitch P^m, meter at this time X/Y

Output: this bass pitch p_i, next bassline direction d_i

1. Get the chord root r_{i_c}, chord bass b_{i_c} and next chord bass b_{i_c+1}, chord start-time t_{i_c}, chord duration Δt_{i_c}

Special bass choices

2. If i_c is the last chord in the piece:
3. $p_i = b_{i_c}$
4. Elif $d_{i-1} =$ "STAY":
5. $p_i = p_{i-1}$

Downbeat-position bass choices

6. Elif $\left|\tau_i - t_{i_c}\right| \bmod 96 = 0$ and $X/Y = 4/4$:
7. $p_i = b_{i_c}$
8. Elif $\left|\tau_i - t_{i_c}\right| \bmod 72 = 0$ and $X/Y = 3/4$:
9. $p_i = b_{i_c}$

First 4th note beat position

10. Elif $\left|\tau_i - t_{i_c}\right| \bmod 96 = 24$ and $\Delta t_{i_c} \geq 24 \times 3$:
11. $p_i = \mathrm{root\_or\_5th}(r_{i_c})$
12. Elif $\left|\tau_i - t_{i_c}\right| \bmod 72 = 24$ and $\Delta t_{i_c} \geq 24 \times 3$:
13. $p_i = \mathrm{root\_or\_5th}(r_{i_c})$
14. Elif $\left|\tau_i - t_{i_c}\right| \bmod 96 = 24$ and $\Delta t_{i_c} = 24 \times 2$:
15. $p_i = \mathrm{leading\_or\_5th}\left(b_{i_c+1}, r_{i_c}\right)$

Second 4th note beat position

16. Elif $\left|\tau_i - t_{i_c}\right| \bmod 96 = 48$ and $X/Y = 4/4$:
17.   If $\mathrm{is\_fifth}\left(p_{i-1} - r_{i_c}\right)$:
18. $p_i = r_{i_c}$
19.   Else:
20. $p_i = \mathrm{5th}\left(r_{i_c}\right)$
21. Elif $\left|\tau_i - t_{i_c}\right| \bmod 72 = 48$ and $X/Y = 3/4$:
22. $p_i = \mathrm{leading\_or\_5th}\left(b_{i_c+1}, r_{i_c}\right)$

Third 4th note beat position (only for 4/4 meter)

23. Elif $\left|\tau_i - t_{i_c}\right| \bmod 96 = 72$ and $X/Y = 4/4$:
24. $p_i = \mathrm{leading\_or\_5th}\left(b_{i_c+1}, r_{i_c}\right)$
25. Else:
26. $p_i = r_{i_c}$
27. Adjust p_i such that (p_{i-1}, p_i) follows the bass direction
28. Adjust p_i such that it stays within $[P^m, P^M]$
29. Determine d_i according to the relationship between (p_{i-1}, p_i) (after adjust)
30. Return p_i, d_i

Implement this function over bass notes, and assemble the list of p_i into the final walking bass.

Overall Algorithm

Algorithm: swingify a piece

Input: melody track $\mathbf{n} = [n_i = (t_i, \Delta t_i, p_i), i = 1, \ldots, N]$, chord track $\mathbf{c}$, meter X/Y

Output: swingified melody track $\mathbf{n}'$, voicing track $\mathbf{v}$, walking bass track $\mathbf{b}$

1. Get swingified melody track: $\mathbf{n}' = \mathrm{swingify}(\mathbf{n})$
2. Incorporate melody notes into chord track: $\mathbf{c} = \mathrm{incorporate\_melody}(\mathbf{n}, \mathbf{c})$
3. Get key detection results from $\mathbf{c}$: $\mathbf{k} = \mathrm{detect\_keys}(\mathbf{c})$

4. Extend the chord track according to the keys: $\mathbf{c}' = \text{extend\_chords}(\mathbf{c}, \mathbf{k})$
5. Use Markov model to determine template of voicings $\mathbf{T_v}$
6. Get voicing track from the extended chords based on the rhythm templates: $\mathbf{v} = \text{get\_voicing\_track}(\mathbf{c}', \mathbf{T_v}, X/Y)$
7. Get walking bass track from the chords: $\mathbf{b} = \text{get\_walking\_bass}(\mathbf{c}', X/Y)$
8. Return $\mathbf{n}', \mathbf{v}, \mathbf{b}$