Music as Flow: A Formal Representation of Hierarchical Processes in Music

Zeng Ren[1], Wulfram Gerstner[1], and Martin A Rohrmeier[1]

[1] École Polytechnique Fédérale de Lausanne

Hierarchical Processes

A time span in tonal music can express a state persisting in time, **a stationary process**, such as a key region or an arpeggiated harmony. It can also signal a change of state, a **transitory process**, such like modulating phase or a descending line. Processes of both kind can give rise to each other. A note as being part of a linear motion, can be tonicized and expanded into a entire phrase, which can in turn give rise to new motions.

To express and encode such semantics from music theory in an elegant way, this paper offers an axiomatic characterization of the temporal organizations of both kind of processes.

Linear Constructions

Construction principles for a **stationary process** on state X and a **transitory process** from state X to state Y.

$$\begin{aligned}\widehat{X} &::= Point\\ &\quad \mid \; Joint\ \widehat{X} \xrightarrow{X \quad X} \widehat{X}\end{aligned}$$

$$\begin{aligned}\xrightarrow{X \quad Y} &::= Unit\\ &\quad \mid \; Link\ \xrightarrow{X \quad Z} \widehat{Z} \xrightarrow{Z \quad Y}\end{aligned}$$

Domain/theory specific semantics such as repetition, passing tone, and neighbor tone elaboration can be modeled using *Joint* and *Link* constructors with the appropriate constraints/predicates on the states involved.

Concurrent Constructions

There are two basic building construction principles to express concurrency of two processes.

$$\widehat{(X,Y)} ::= \widehat{X} \infty \widehat{Y}$$

$$\xrightarrow{(X,X') \qquad (Y,Y')} ::= \xrightarrow{X \quad Y} \Uparrow \xrightarrow{X' \quad Y'}$$

When, denoted as ∞, express simultaneity of two stationary processes. **While**, denoted as $\Uparrow$ expresses concurrency of two transitory processes.

These two constructions expresses two independent processes of the same status as a whole.

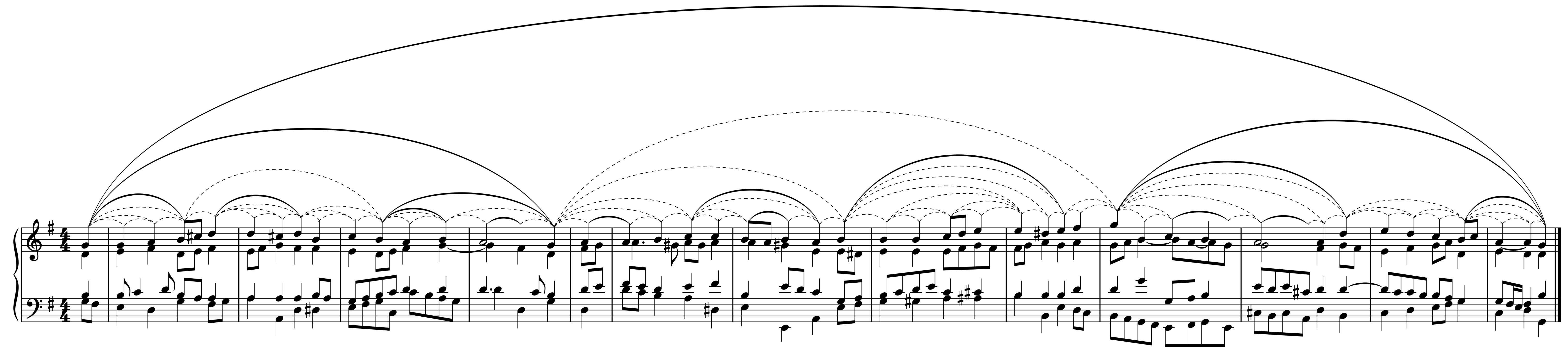


Figure: An analysis of Bach Chorale No 9 BWV 248 using the proposed framework. A solid slur contains a dashed slur in between two solid slurs represents the *Joint* constructor. A dashed slur containing a solid slur in between two dashed slurs represents the *Link* constructor. A *Point* is represented as an attaching point of the solid slur, just like a boundary point of a closed interval. A *Unit* is represented as a dashed slur containing nothing, analogous to an open interval.

Algebraic Derivations

Passing-neighbor elaboration:

$$\begin{matrix} 3-2-1 \\ 1-7-1 \end{matrix}$$

$$\widehat{(3,1)}\Big(\big(\xrightarrow{3\ 2}\Uparrow\xrightarrow{1\ 7}\big)\widehat{(2,7)}\big(\xrightarrow{2\ 1}\Uparrow\xrightarrow{7\ 1}\big)\Big)\widehat{(1,1)}$$

2-against-1 passing elaboration:

$$\begin{matrix} 5-4-3 \\ 7\quad -\quad 1 \end{matrix}$$

$$\widehat{(5,7)}\Big(\big(\xrightarrow{5\ 4}\widehat{4}\xrightarrow{4\ 3}\big)\Uparrow\big(\xrightarrow{7\ 1}\big)\Big)\widehat{(3,1)}$$

Suspended passing elaboration:

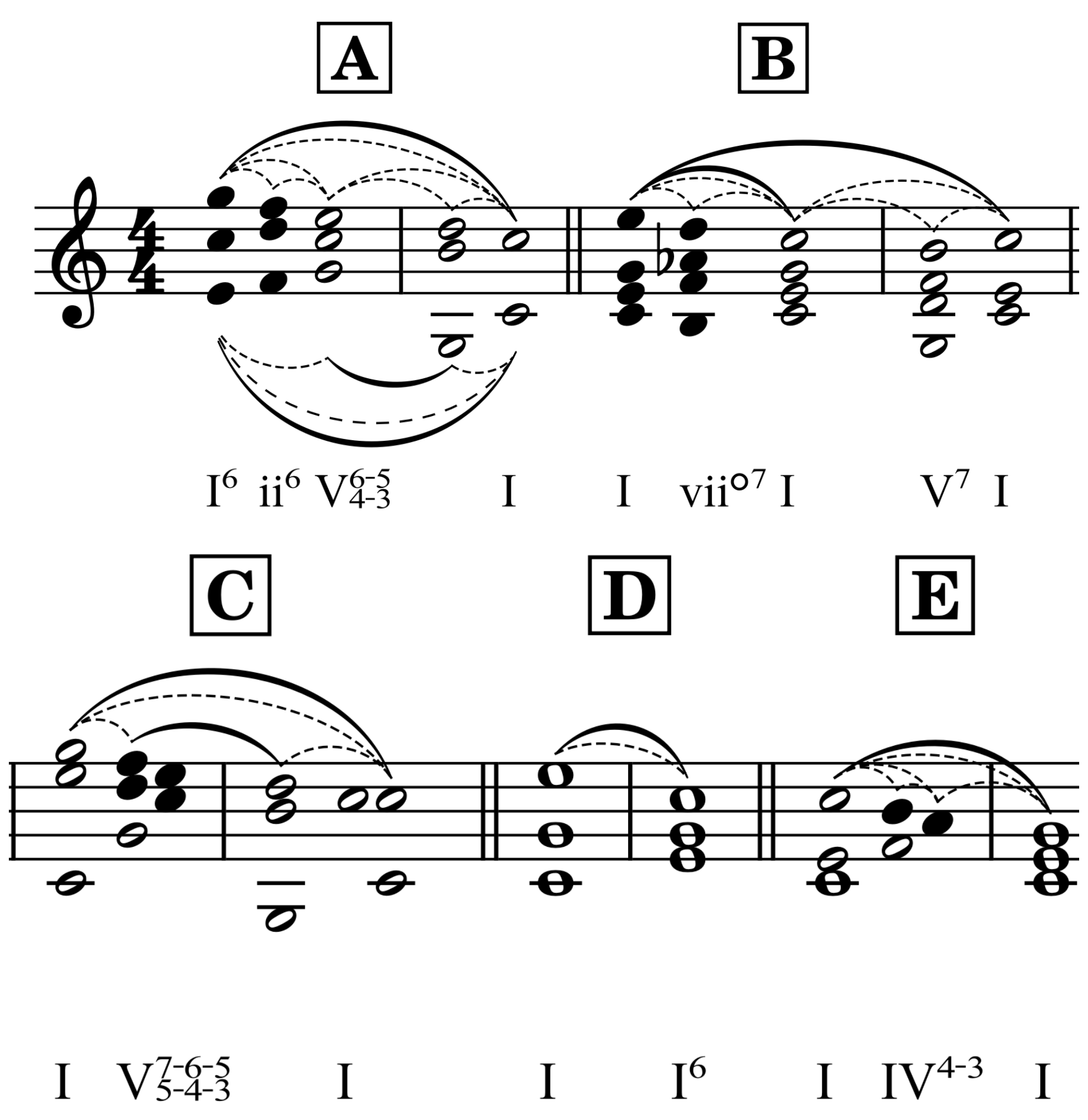
$$\begin{matrix} \overset{\frown}{3 \quad 3}-\overset{\frown}{2 \quad 2}-1 \\ 5-\overset{\frown}{4 \quad 4}-\overset{\frown}{3 \quad 3} \end{matrix}$$

$$\widehat{(3,5)}\Big(\big(\xrightarrow{5\ 4}\uparrow_{\uparrow}\xrightarrow{3\ 2}\big)\widehat{(2,4)}\big(\xrightarrow{4\ 3}\uparrow_{\uparrow}\xrightarrow{2\ 1}\big)\Big)\widehat{(1,3)}$$

where

$$\xrightarrow{x\ y}\uparrow_{\uparrow}\xrightarrow{x'\ y'}=\xrightarrow{(x,x')\ (y,x')}\widehat{(y,x')}\xrightarrow{(y,x')\ (y,y')}$$

Graphical Notation



The mutual nesting of stationary and transitory processes are visualized using the nesting of two type of slurs. Solid slurs represents stationary processes while dashed slurs represents transitory processes.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 760081 – PMSB. We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology.
We thank the team of the Digital and Cognitive Musicology Lab (DCML), particularly Gabriele Cecchetti and Xinyi Guan, for their helpful comments on earlier versions of this paper. Furthermore, we thank Yannis Rammos and Christoph Finkensiep for fruitful discussions on the mu- sic interpretations and implementation of the model in this paper.