Harmonic Syntax in Webern's Opp. 9, 10, and 11

A Digital Approach

Joshua Ballance

King's College, Cambridge

This dissertation is submitted for the degree of Master of Philosophy.

Abstract

This project examines harmonic syntax in three pre-serial atonal works by Anton von Webern: *Sechs Bagatellen*, Op. 9, *Fünf Stücke*, Op. 10, and *Drei Kleine Stücke*, Op. 11. These three pieces, comprising 14 movements, and approximately 12 minutes of music, are effective exemplars of Webern's early atonal style across three genres (respectively, string quartet, orchestra, and cello & piano). They were written in a narrow chronological span, 1911-1914, and as such offer a perspective on his expressive language at this time.

The methodological approach employed in this project is the use of computer code to amass data representing the number and frequency of the simultaneities and simultaneity types that Webern uses, and the various sequences in which he employs these. Simultaneities present a very tangible and aurally immediate part of these works: they are some of the most overtly perceptible expressions of harmony. Using this method thus provides an accurate sense of Webern's practice across the works: rather than extrapolating from limited exemplars, it is possible to garner a real understanding of his language.

This statistical approach is essentially new to analysis of music from this period, even if Jackson (1970) had attempted to do something similar almost half a century ago. It is thus a fundamentally interdisciplinary project, straddling the intersection of computational musicology and formal analysis. In addition to the digital methodological innovations, the comparative nature of the study is novel: by comparing harmonic units across three works from a compressed chronological span, it is possible to assess the extent to which Webern's 'freely' atonal music displays a consistent harmonic grammar.

The principal finding of the project is quantifying the sheer heterogeneity of his harmony. In almost every domain Webern avoids repetition and the ensuing creation of hierarchies. Nonetheless, detailed examination of the results exposes a more textured picture of the situation, indicating disparities in his practice both between and within the works.

Table of Contents

List of Examples

List of Figures

List of Tables

Introduction

Pitch material has long been a preoccupation of analysts, and the early post-tonal music of the Second Viennese School has undergone significant appraisal in this area. In particular, there has often been an ambition to assess 'common practice' – that behaviour most typical of composers in a given period. With the advent of modern digital technologies, analysts are presented with a new set of tools with which to engage in these important enquiries. Perhaps most significantly, the speed and flexibility of contemporary computers allow routine data gathering tasks to be accomplished swiftly and easily, facilitating the compilation of large datasets about given works.

This project assesses Webern's early post-tonal compositional output by examining three works from a close chronological span but from different genres, to provide a perspective on his expressive voice at this time. These works are his *Sechs Bagatellen für Streichquartett*, Op. 9 (1911-1913), *Fünf Stücke für Orchester*, Op. 10 (1911-1913), and *Drei Kleine Stücke für Violoncello und Klavier*, Op. 11 (1914).[1] Collecting data digitally enables this project to consider the entirety of these pieces, rather than isolating individual passages to function as a synecdoche for broader practice, a typical analytical approach. The project is therefore interdisciplinary, with a hybrid of digital and traditional tools. The focus is on pitch material, which is considered from the perspective of vertical simultaneities.[2] As such, there are two primary research questions: 1) Are there particular types of vertical pitch-simultaneity that are used more than others? And 2) Are there sequences of simultaneities – progressions – that are used more than others? These two questions constitute a broader enquiry: was Webern attempting to establish a new harmonic "vocabulary", and what were the implications for "syntactical" patterns?

This thesis is organised according to the following structure. After situating the project within the fields of both digital musicology and post-tonal analysis, there is an exposition of the methodology used in the project. This primarily includes defining and categorising the types of data accumulated in the project, and outlining the digital approach used for the data gathering. The resulting data are then presented and interrogated with regard to the research questions cited above. Finally, *Sechs Bagatellen* (SBFS) is considered in the light of the work already done. Framed in the context of previous analysis of this piece, this case study seeks to demonstrate some of the possible advantages that this digital approach might provide, such as the interrogation of earlier analytical ideas.

---

[1] Henceforth, these will be referred to respectively as SBFS, FSFO, and DKS.

[2] Throughout this thesis, 'simultaneity' is refers to any vertical collection of pitches, irrespective of the number of elements it contains. Although this taxonomy is imperfect, it is preferable to 'sonority' (implying timbral information), 'chord' (including multiple pitches), or 'harmony' (categorically suggesting a functional relevance).

Chapter 1: Literature Review

Digital Musicology

With its commitment to a digital process for data collection, this project is clearly situated within the broad arena of digital musicology, a field which has been operating primarily for the last half-century, alongside the general rise of the digital humanities. Whilst it is beyond the scope and ambition of this thesis comprehensively survey the scope of digital or computational musicology, the review subdivides the discipline in order to locate the present work within the larger field.

Within this area, the present work fits into the domain of Music Information Retrieval (MIR), in which digital methods are used to extract certain features from some musical source. Stephen Downie defined MIR as 'a multidisciplinary research endeavor[1] that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world's vast store of music accessible to all.' (Downie, 2004, p. 12) He further subdivided the field into three principal areas of research: 1) 'Query' systems, which provide users with something akin to a musical 'search engine'; 2) Music recommendation and distribution systems; and 3) Music analysis systems. Although there has been successful research in all of these areas over the last 15 years, these subdivisions continue to indicate the main fields of enquiry.

It is to the third of these research areas, music analysis systems, that this study belongs. This domain is typically subdivided according to the source material employed by the analyst, either audio files (typically recordings) or musical scores as the 'music' from which to retrieve information. This project uses scores as the source of information about the music. This approach in itself has an extensive history; nonetheless, the matter of encoding scores has remained an overriding concern for decades. Even a brief survey over the last half-century of research displays a wide sweep of alternative approaches. Ramon Fuller (1970) described his idiosyncratic method of encoding some work by Webern in FORTRAN, a contemporary coding language; six years later, Hans Janssens and Walter Landrieu described the Melowriter, a machine they developed in Ghent which 'makes it easy for the user to codify musical scores' (Janssens & Landrieu, 1976, p. 255). Six years after that, Giovanni B. Debiasi and Giovanni G. de Poli posited another approach, outlining the MUSICA language for encoding musical scores, which apparently satisfied 'all the descriptive and functional requirements of instrumental music' (Debiasi & de Poli, 1982, p. 2). A decade later in 1993 Marcel Mesnage described a software system he called the 'Morphoscope', which sought to 'build a formal model of the complete score' (Mesnage, 1993, p. 119). From this plethora of different approaches, it is clear that a single

[1] American English quotations have been preserved throughout, although the thesis uses British English.

universally-applicable approach had failed to materialise. Likewise, in many of these cases the focus for the authors was often the system of encoding and data collection, rather than the possible ramifications these tools could have for the understanding of music. Despite extensive activity, the fragmentation of the field thus resulted in little progress. As Nicholas Cook described such activity: 'a sustained burst of initial enthusiasm is followed by running out of money, resulting in software that is sometimes less than fully functional, often less than fully documented, rarely properly supported, and usually soon obsolete' (Cook, 2004, p. 107).

The situation over the last two decades has been somewhat more productive and can be generally subdivided into two principal areas. On the one hand there are projects that seek to improve the universal database of encoded scores available for digital musicologists to work with; on the other, there are various sets of tools available for conducting these enquiries. As to the first of these, although there remains inevitable institutional separation, the widespread adoption of the internet and the collapse in financial costs surrounding computing has allowed for a greater centralisation of work. The most significant remaining division here concerns the plethora of different filetypes in use. Although XML/MXL has become something of a standard, there are various other formats in use, including Humdrum, Musedata, Lilypond, MEI, and others. In a 2015 call for the widespread adoption of MEI, Laurent Pugin pointed out that 'The development of music computer codes has shown us how different centers of interest and different focuses can lead to countless barely compatible initiatives', and he argued that the MEI project is 'well placed to play a unifying role' (Pugin, 2015, para. 8). Nonetheless, conversion is often feasible, allowing different scholars to work in their own preferred environment. Alongside these different formats, various different sites host different collections of repertoire. Whilst some, such as IMSLP (Guo, n.d.), MuseScore (Bonte, Froment, & Schweer, n.d.), or ELVIS ("ELVIS Project," n.d.), aspire to universal coverage, others like the Josquin Research Project (Rodin, Sapp, & Bokulich, n.d.) or the Lieder Corpus Project (Rootham, Jonas, & Gotham, n.d.) are more specialised.

This project employs music21 as the primary analytical tool. This is a Python-based toolkit for enabling digital musicology (Cuthbert & Ariza, 2010). music21 provides a library of tools with which the analyst can write code to be applied to encoded scores, resulting in the extraction of data concerning features of the score. A wide variety of filetypes can be considered by music21, and its flexibility allows for a broad range of types of enquiry. Meanwhile, because it operates within the broader Python language, data processing achieved through "conventional" coding can be integrated into the same environment. The ELVIS team have used music21 to produce the VIS Framework for Music Analysis. Though this is not used in the present project, it is an interesting example of the gradual spread of programming systems; indeed, according to the authors, the aim of the framework is 'to lower the barrier to empirical music analysis' ("VIS Framework for Music Analysis," n.d.).

Corpus Study

Having positioned this project within the broader field of digital musicology, the related approach of 'corpus study' requires some elucidation. This is a research strategy derived from linguistics, with four main tenets: it is an empirical approach, analysing patterns of use; it employs a large collection of texts ('the corpus') from which to derive these patterns; it makes use of computers for analysis; it utilises both quantitative and qualitative techniques (Biber, Reppen, & Friginal, 2012, para. 2). As an approach in musicology, it has proved relevant to a large variety of interests, from assessing strategies of tonal harmony in Bach chorales (Rohrmeier & Cross, 2008), to characteristics of jazz solos (Weiß, Balke, Abeßer, & Müller, 2018). In the case of this study, the corpus consists of the three works by Webern; the patterns of use regard his pitch-based compositional tendencies. As such, this may be considered a small corpus, as typically corpus studies utilise a multitude of pieces, allowing patterns to be revealed from a large body of material. Martin Rohrmeier & Ian Cross's study, for example, uses a corpus of 386 Bach chorales; Weiß et al. consider 456 jazz solos. Indeed, one of the advantages of digital methods is the ability to survey a broad range of sources, without the conventional constraints of time or accuracy that would be imposed upon an analyst carrying out such a study manually. Indeed, Cook argues that this comparative quality is the primary advantage of computational data collection-based approaches (Cook, 2004, pp. 107–109). Nonetheless, the small corpus of works used here generate an enormous quantity of data, making a corpus study-style strategy indispensable. As an example, in just these three works 662 different simultaneities were identified, putting the study beyond the practicable realm of manual identification and processing.

Corpus studies, and computational musicology more broadly, have typically focussed on works from the common practice period rather than the twentieth century. In part this is due to the availability of encoded scores: copyright restrictions have encouraged musicologists to encode scores that are out of copyright, to allow them to be shared freely online; conversely, authors seeking to work with already-encoded scores are typically limited to pre-twentieth-century ones. This is primarily because bodies of work in the twentieth century that can be understood as meaningfully comprising a stylistically homogeneous corpus are rarer, largely due to the increased fragmentation of twentieth-century aesthetics. As for relating research to previous theories, the sheer quantity of work that has been carried out on canonical tonal works has presumably encouraged computational musicologists to focus on these bastions of the canon, in order to associate their own work with this significance. All of these factors explain the common use of Bach's chorales as a corpus, for example (Jacoby, Tishby, & Tymoczko, 2015; Rohrmeier & Cross, 2008; White, 2013). As such, whilst the three works that form the basis of this thesis form an effective and useful corpus, there has been little analysis of this type undertaken with anything from this period.

Post-Tonal Analysis & Webern Studies

The other dimension of this project is its position within the field of post-tonal analysis, and more particularly the analytical work that has been carried out on the Second Viennese School and these pieces. There are various strands of analytical enquiry which pertain to this work: most obviously the Pitch-Class Set Theory of Allen Forte and his disciples, but also neo-Schenkerian theories, and debates over segmentation and similarity relations. A selection of these scholars also engage directly with the three works under examination here, either foregrounding them in their investigations, or using them as case studies for their theory (e.g. Chrisman, 1979; Davies, 2007; Forte, 1998; Lewin, 2010).

Forte's Pitch-Class Set Theory, as initially outlined in the seminal *Structure of Atonal Music* (Forte, 1973) and extended by other authors, for example Larry Solomon (1982), is the dominant theory in the analysis of early post-tonal music, of which Webern's Opp. 9, 10, and 11 are exemplars. Indeed, authors like David Lewin (1983, 2010) have applied a set-theoretic approach to some of these pieces, and general aspects of the theory certainly influence this project. The methodology below will indicate points of departure from Forte: though the conceptual foundations are useful, the theoretical details and the more complex augmentations have been avoided. This is a fairly typical approach in contemporary analysis: Forte's formalisation of pitch class sets as the basis for a structural understanding of a work is widely-accepted and is used uncontroversially in much consideration of this music. Nonetheless, aside from the more fundamentalist school of neo-Fortean analysis, the broader scholarly field uses these basic concepts as local tools in a search for alternative approaches (e.g. Davies, 2007; Roig-Francoli, 2001). A full critique of Forte's approach is beyond the scope of this essay, as is a survey of the notable polemic surrounding his work (e.g. Dipert, 1977; Taruskin, 1979), or Forte and Taruskin's famous correspondence (Forte, 1986; Taruskin, 1986).

The debate surrounding segmentation is a topic with significant pertinence to the present discussion. The essential question here is what constitutes a unit of harmonic significance. Christopher Hasty has outlined many of the basic ideas in this discussion. The fundamental problem in post-tonal analysis, as he puts it, is as follows: 'any interval is capable of being heard as self-sufficient; thus, in principle, any pitch may be associated with any other pitch and any number of pitches may conceivably be heard sounding together (*con-sonans*) as a comprehensible harmonic unit' (Hasty, 1981, p. 55). Edward Pearsall similarly argues that 'chords do not necessarily represent harmonic units' (Pearsall, 1991, p. 348). For Forte, segmentation is 'virtually impossible to systematize' (Forte, 1973, p. 91). Though he acknowledges that there are some segments which are 'isolated as a unit by conventional means' (Forte, 1973, p. 83), the major difficulty comes with those segments which are less overtly demarcated. For discerning these segments, Forte suggests considering '*contextual* criteria', which he defines as 'references to the local context of the candidate segment or […] non-local sections of the music' (Forte, 1973, p. 91). This is similar to Hasty's idea of "domains", which are the properties (pitch, timbre, dynamic, etc.) of a given musical element (Hasty, 1981, p. 57). According to Hasty, domains are "transparent" of each other, and so to define a distinct new musical element there must be a "discontinuity" in at least one domain, though the others may be unchanged. Segmentation is thus carried out according to the cumulative effect of discontinuities in enough domains: it is 'the formation of boundaries of continuity and discontinuity which result from the structures of various domains' (Hasty, 1981, p. 59). For Forte, there is a further method for discerning segmentation which uses the pitch class content of the segments themselves. He proposes four criteria:

> (1) the set occurs consistently throughout—it is not merely "local"; (2) the complement of the set occurs consistently throughout; (3) if the set is a member of a Z-pair, the other member also occurs; (4) the set is an "atonal" set, not a set that would occur in a tonal work. (Forte, 2006, p. 45)

These betray a common critique of Forte's method, however. Using the pitch class content of the sets themselves to validate their own segmentation is precisely what provokes the common critique that the segments are chosen in order to create a neat analytical outcome – the end justifying, well, the end.

Similarity is another important topic within post-tonal analysis. Here Forte again has much to say, staking out two main areas: equivalence and similarity. For Forte, two sets are equivalent 'if and only if they are reducible to the same prime form by transposition or by inversion followed by transposition' (Forte, 1973, p. 5 see pp. 5-11 for a broader exposition). Of course, implicit in this is the reduction involved in expressing a set in its normal order. Further, Forte goes on to define similarity relations in certain domains, in order to ascertain the 'degree of similarity' between two sets (Forte, 1973, p. 46). A comprehensive exposition of his theory is beyond the scope of this review, suffice to summarise the four relations to which he draws attention: R_p ('Maximum similarity with respect to pitch class'); R_0 ('Minimum similarity with respect to pitch class'); R_1 ('Maximum similarity with respect to interval class'); R_2 ('Minimum similarity with respect to interval class') (Forte, 1973, p. 49 see pp. 46-60 for a full survey). Used together, Forte views these basic properties as helpful indicators of degrees of similarity. Two related concepts are those of the inclusion relation, the existence of subsets within a superset, and invariance, that subset common to two non-equivalent supersets (Forte, 1973, pp. 26–46). Miguel Roig-Francoli has sought to build on these ideas in his theory of Pitch-Class-Set Extension (PCSE). For him, two sets are connected if they 'have at least one pitch class in common (common-tone connection; the degree of connectedness will be given by the number of common tones: the higher the degree, the stronger the connection), or if at least two of their respective pitch classes are related by IC1 (chromatic connection)' (Roig-Francoli, 2001, p. 64 see 64-70 for the full theory). He thus provides a more variable way of comparing similarity relations than Forte.

Roig-Francoli's discussion of set connections forms part of his larger discussion about the potential for 'extension', which itself is part of the general debate surrounding prolongation. For Joseph Straus, 'post-tonal music is not prolongational or, to put it another way, prolongation as an analytical tool will not produce significant results' (Straus, 2006, p. 7). In his view, there are four conditions required for effective prolongation in a neo-Schenkerian sense, and though it might be theoretically possible for post-tonal composers to achieve these, in practice he posits that they have not. His alternative is to argue for an 'associative' theory for the structural middleground, which allows for connections between two non-consecutive elements, but not the prolongation of the first to the second through the intervening elements (Straus, 2006, pp. 13–15). Pearsall (1991) disagrees, however, proposing that all four of Straus's conditions can be achieved on a local level, which would thus generate a piece's own patterns for satisfying these requirements. Returning to Roig-Francoli, though he posits that a neo-Schenkerian background is unachievable, he theorises that Pitch-Class-Set Extension allows for prolongation on both foreground and middleground levels, creating a Pitch-Class-Set Extension Region which is influenced by the governing set (Roig-Francoli, 2001, pp. 71–82).

Further Work

Clearly there has already been plenty of analytical work carried out on pieces from this period, and so a justification of the utility of similar work is required. An empirical perspective allows those using these methods to consider some of the theory that is often advanced in a more abstracted sense, in the context of 'real' music. Applying these data gathering techniques to this repertoire allows for a novel perspective on this music and, as will be shown below, a refinement of some of the analytical processes that have previously been put forward. More broadly, conducting digital analysis of post-tonal work in this way allows the development of tools that will be useful for a broader range of repertoire. As the code written for this project will be shared under open-source parameters online (see Butterfield & Ekembe Ngondi, 2016), it will be easily available for other scholars, who could adapt it in order to apply it to a broad range of repertoire.

Chapter 2: Methodology

Introduction

This chapter surveys the methodology employed in this project for collecting the data and defines precisely what the different data represent. Initially, the chapter discusses the process of encoding the scores to provide the source-material, before moving on to the intricacies of the data collection.

Encoding

The encoding of these scores was done manually; although various optical music recognition tools were trialled, the small size of the corpus and the relative sparsity of each piece meant that manual encoding was more efficient. The pieces were encoded exactly as they appear in the original Universal Edition scores (Webern, 1923, 1924b, 1924a). This initial act of encoding required little editorial intervention; the only area in which interpretation might have been required was the precise positioning of tempo changes, but in practice these tend to be positioned above identifiable gestures, implying their location.

In order to produce files which were appropriate for music21 to work with, however, various alterations had to be made. Trills and fast alternations between notes were treated as static prolonged notes, rather than creating new attacks (see Example 2-1 for an example of an original passage and Example 2-2 for the altered version). In the case of trills, this means that the upper note is treated as fundamentally decorative, and so harmonically non-essential; as for fast alternations, these are interpreted as essentially sounding a dyad[1], rather than repeating the two notes. Another pitch-related alteration was the conversion of all artificial harmonics to sounding pitch (to the nearest semitone), rather than their notated form (see Example 2-3 and Example 2-4). Arpeggiated chords were interpreted as constituting one attack, and it was assumed that the piano sustaining pedal was not used to extend pitches beyond their notated value. Unpitched percussion were removed, and it was also assumed that any timbral distortions (e.g. *sul pont.*) has no effect on pitch content.

Whilst these decisions all require editorial judgement, they are mostly intuitive. Those that are not are the inevitable effects of working with a score as source material rather than audio recordings. It is possible (though it seems unlikely given the ease with which the music falls under the hand) that a pianist performing DKS might use the sustaining pedal, but there is no way to predictively model for

[1] In this thesis, the cardinality of simultaneities will be referred to using terminology ending in -ad (e.g. monad, dyad, etc.). The use of the term 'triad' thus refers simply to a 3-note unit, with no tonal harmonic implications.

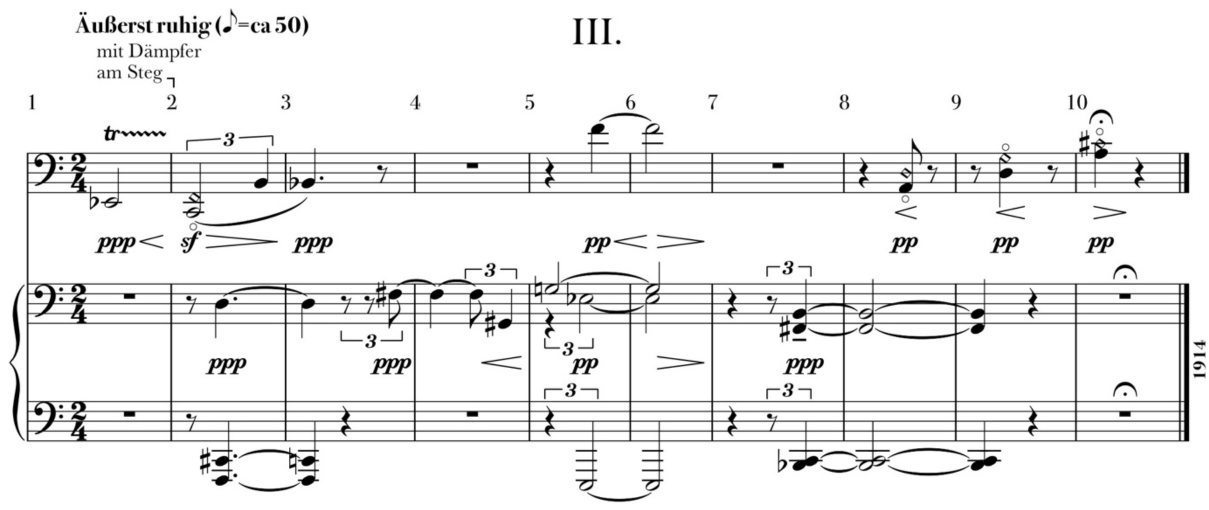
this. Likewise, it is plausible that *sul pont.* string notes might imply a fundamental pitch of the second partial, rather than the first (i.e. an octave higher than notated). Again, there is no reliable way of anticipating this, and so no way of adjusting for it.

The other major modelling assumption involves tempo. There are metronome indications throughout all of these pieces, and so although performers may well differ from these, they have been employed strictly. Regarding tempo changes, these are modelled as either a *ritardando* or an *accelerando* (*zögernd* has been treated as the former; *drängend* as the latter). These, and *fermatas*, are calculated according to the playback of Sibelius 7. Although the result is inevitably 'wooden', it suggests how the tempo might change according to the indications given by Webern.

Example 2-1: FSFO, Movt. III, bb.1-3, original version

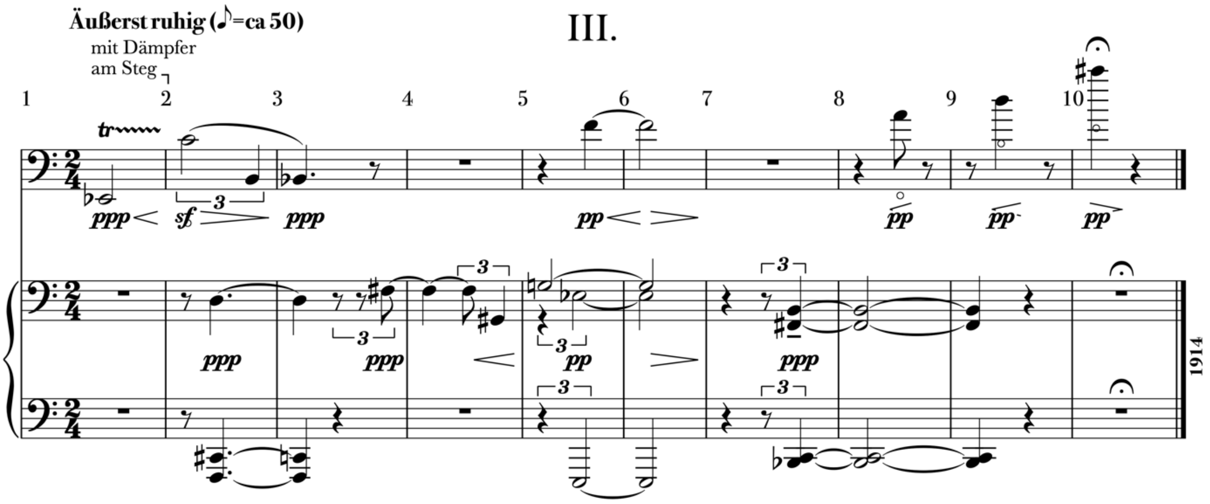
Anton Webern, '5 Stücke für Orchester, Op. 10' © 1923, 1951 by Universal Edition A.G., Wien/UE5967

Example 2-2: FSFO, Movt. III, bb.1-3, sounding version



Example 2-3: DKS, Movt. III, original version

Anton Webern, '3 Kleine Stücke für Violoncello und Klavier, Op. 11' © 1924, 1952 by Universal Edition A.G., Wien/UE7577



Example 2-4: DKS, Movt. III, sounding version

Data Collection: Concepts

The approach employed for the data collection was to create a list of all vertical simultaneities in each of these pieces, and then record them in various forms (Table 2-1). For each type of single simultaneity and each pitch, the total duration is given in seconds; for each sequence of simultaneities, the number of statements of the sequence is provided, as this is a more appropriate measure of the significance of the sequence. Before discussing the details of this process of data collection, however, some conceptual decisions require explication.

There is clearly a foundational premise here that the greater the frequency (respectively as a proportion of a work, or in the number of statements) of a given element, the greater its significance. Simultaneities can, of course, attain significance in a work in other ways: orchestration, dynamic, metre, all have a place to play. Nonetheless, as expressed above, the fundamental aim of this project is to assess whether Webern was developing a new set of simultaneities, distinct from that of tonal practice. As such, the frequency of these simultaneities is an important part of considering their significance. This sort of work has an analogy in tonal practice, where several authors have considered frequency of different elements as an indication of significance. Jason Yust (2019) has done so in relation to pitch class content in a corpus of common practice period material; Rodolfo Moreno (2017) and Rohrmeier & Cross (2008), meanwhile, have considered harmonic progressions in Bach's chorales. In all three cases, frequency of appearance is understood to be a legitimate indicator of significance, a conclusion which, after all, makes intuitive sense.

Type	Definition	Example
Pitch Complex	Untransposed simultaneity recorded in absolute pitches.	(A3 D4 F#4 G#4)
Class Complex	Transposed simultaneities (to bass-note C4) recorded in absolute pitches.	(C4 F5 A6 B6)
Class Collection	Transposed simultaneities (to bass-note C4) recorded in pitch classes (ascending order).	(0 5 9 11)
Pitch Complex Sequence	Sequence of two and three successive pitch complexes.	(E2 F3 B4) (A3 D4 F#4 G#4)
Class Complex Sequence	Sequence of two and three successive class complexes.	(C4 C#5 F#5) (C4 F5 A6 B6)
Class Collection Sequence	Sequence of two and three successive class collections.	(0 1 6) (0 5 9 11)
Absolute Pitch	Individual notes recorded in absolute pitches.	(A3)
Pitch Class	Individual notes recorded in pitch classes (0=C).	(9)

Table 2-1: Types of Data Recorded

The decision to consider vertical simultaneities in this way is also an important analytical decision: Pearsall argued that ‘chords do not necessarily represent harmonic units’ (Pearsall, 1991, p. 348), and Hasty suggested that ‘any pitch may be associated with any other pitch and any number of pitches may conceivably be heard … as a comprehensible harmonic unit’ (Hasty, 1981, p. 55). Segmentation is thus a crucial part of the methodological decision-making process. As discussed above in relation to Forte’s practice, a common critique is the subjective selection of segments in order to provide a satisfying analytical result. To avoid this potential pitfall and provide a comparable and rigorous approach across all three pieces, this analysis applies the same technique to each work, taking account of a crucial part of the musical surface.

As for those types of data recorded, by ‘reducing’ the information in a variety of ways, and considering these different data types, the analysis provides several ways of looking at Webern’s practice. The pitch complex clearly retains the most information, considering both pitch position and voicing and octave doubling; the class complex and class collection instead consider different ‘types’ of simultaneity, the former with regard to voicing and doubling, the latter without. It is worth noting that none of these is identical to either Forte’s “prime form”, or “normal order”, although the basic idea of reduction is the same (Forte, 1973, pp. 3–5). As for spelling, this is preserved in the pitch and class complex, but not in the class collection: thus, (E G# B) and (E Ab B) are recorded as different pitch and class complexes, but the same class collection.

Regarding the sequences of these simultaneities, two- and three-element sequences have been recorded. This decision was, in part, made on the basis of some preliminary results: so few sequences with this many elements are used more than once that to record longer sequences would largely be uninformative. From a more theoretical standpoint, restricting consideration to small-scale patterns like these allows for meaningful enquiry regarding local syntax. According to Moreno, the principle of harmonic syntax refers to 'the norms of precise logical order in the succession of harmonic functions or chords in any harmonic progression in tonal music' (Moreno, 2017, para. 1); Rohrmeier & Cross are rather more circumspect, arguing that 'It is not claimed at all that these mere statistical features constitute harmonic syntax. These statistical features may rather indicate the existence of some underlying features of syntactical organisation of harmonic structure' (Rohrmeier & Cross, 2008, para. 42). Whichever definition of syntax is preferred, the relevance of patterns of small-scale progressions is evident. Thus, whilst this project does not seek to explain any large-scale features of Webern's harmonic syntax, it provides insight into his short-term tendencies.

The final data types recorded are pitches and pitch-classes, irrespective of the simultaneities in which they occur. This gives an indication respectively of the spread of register that Webern uses, and whether there are any hierarchical patterns in relation to pitch-class distribution in these pieces.

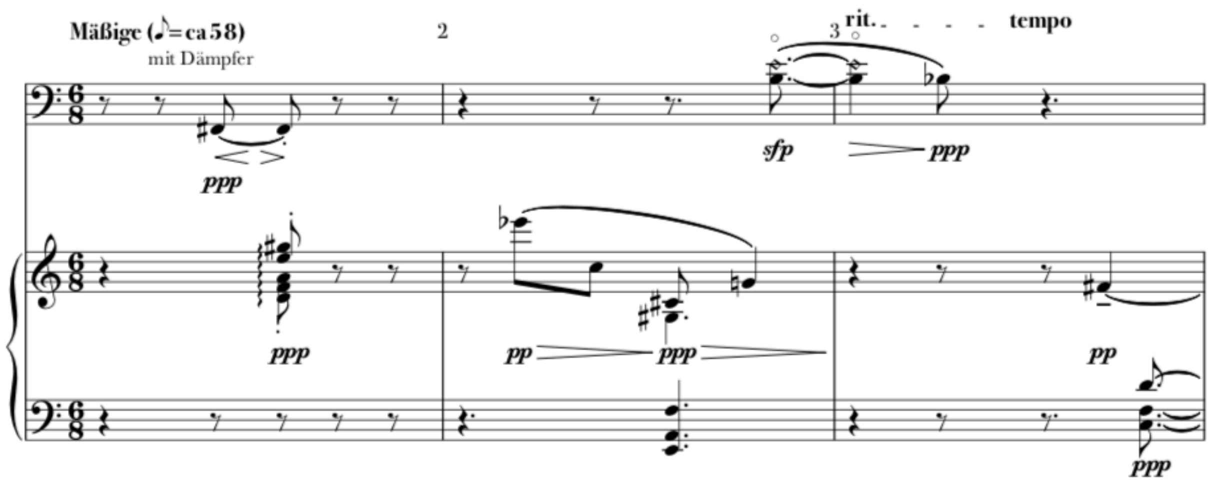
Data Collection: Process

The procedure for collating this data is conceptually simple. It involves creating a list of all vertical simultaneities in each of these pieces, and then recording them in the various forms expressed above. The code for doing so is given in Appendix 1. The following description of the methodology explains the process in some detail, including an explanation for various adjustments required to achieve such an aim.

For the simultaneities, both individually and as sequences, the first step is to "chordify" the piece. This technique takes a given passage and reduces it 'to a series of chords representing the music sounding at each moment in the score' (Cuthbert, Hadley, Johnson, & Reyes, 2012, para. 11). An example is given below: Example 2-5 presents bb.1-3 of Webern's Op. 11; Example 2-6 presents that same passage having been chordified. Although the result is visually abrasive, and ignores matters of orchestration, dynamic, and voice-leading, the information that it provides is invaluable, indicating the total pitch content at any given time. From this total list of all the simultaneities in the piece, a second list is compiled of every different *type* of simultaneity. The total duration of each of these types of simultaneity across the entire piece is then recorded, providing the final list of data. In the case of class complexes and class collections, the simultaneities are transposed to bass-note C4 prior to the compilation of the list of simultaneity-types.

The process is essentially the same for the sequences of simultaneities, but with one caveat, due to the measurement in statements rather than durations. If a given simultaneity is notated by being tied to a consecutive notation of the same simultaneity (e.g. across a barline), it is crucial to know that the second statement is not a new attack of the same simultaneity, but an extension of the first. Therefore, after having chordified the work, all simultaneities that are preceded by the same pitch material tied to it are removed from the list. Following this, the list of all sequences of simultaneities in the piece is compiled, and the number of statements recorded as above.

In categorising pitches and pitch classes, the procedure is even more simple: a list is compiled of the total set of pitches and pitch classes used, and then the total duration of each is recorded. Regarding spelling, the policy is as above: for pitches, spelling is preserved (i.e. G# ≠ Ab); for pitch classes, obviously it is not (8 = 8).



Example 2-5: DKS, Movt. I, bb.1-3

Example 2-6: DKS, Movt. I, bb.1-3 chordified

Chapter 3: Results

Introduction

This chapter comprises an overview of the data collected in this project, with some brief statistical conclusions about the data, both individually for each piece and comparatively. The 'musical' implications of these comments are considered more fully in the following chapter. The full results are in Appendix 2. In order to consider the different aspects of the music, this chapter will outline the results in a number of ways. Initially the patterns regarding pitches and pitch classes will be discussed before simultaneities themselves and then sequences. For the latter two categories, the overall situation for different levels of harmonic reduction will be displayed, before more focussed enquiry regarding particular features of the data.

Regarding the statistical methods employed, the same approaches have been applied throughout the project in order to give simple comparison between categories of data. As the data are so varied, nonparametric methods have been used to analyse them. To identify those elements used significantly frequently, 'point outliers' have been sought, categorised as "novelties". In this categorisation, Tukey's rule has been employed: that outliers are defined as values greater than 1.5 multiplied by the interquartile range, from each quartile. Durations have typically been converted into percentages to allow for meaningful comparisons between the pieces.

Pitches

The data for pitches and pitch classes are conceptually simplest and provide some interesting comments on both the pitch content and the texture of the pieces. Figure 3-1 provides the pitch data for all three pieces, with C2, C4, and C6 marked for scale. Enharmonic equivalents have been treated as the same pitch (so G# = Ab). To clarify precisely what these data are, the value for each pitch is the duration of that pitch as a percentage of the total duration of all pitches in the piece (which is not the same as the total duration of the piece, which would include silences). All three pieces have a fairly similar span of total pitches used (DKS: 52; SBFS: 63; FSFO: 57).

Regarding novelties, DKS & SBFS have two each: respectively G3 & F#2, and F#4 & E4. For FSFO, there are seven novelties: E5, C#6, C4, G#3, D6, D5, & A#4. As for the interquartile range (IQR), these are: DKS: 2.1; FSFO: 1.4; SBFS: 2.0. The IQR indicates the spread of the data, thus showing that whilst the pitches in DKS and SBFS are very similarly spread, FSFO is marginally more centrally concentrated. Nonetheless, as is clear from Figure 3-1, DKS is concentrated lower in pitch than SBFS.

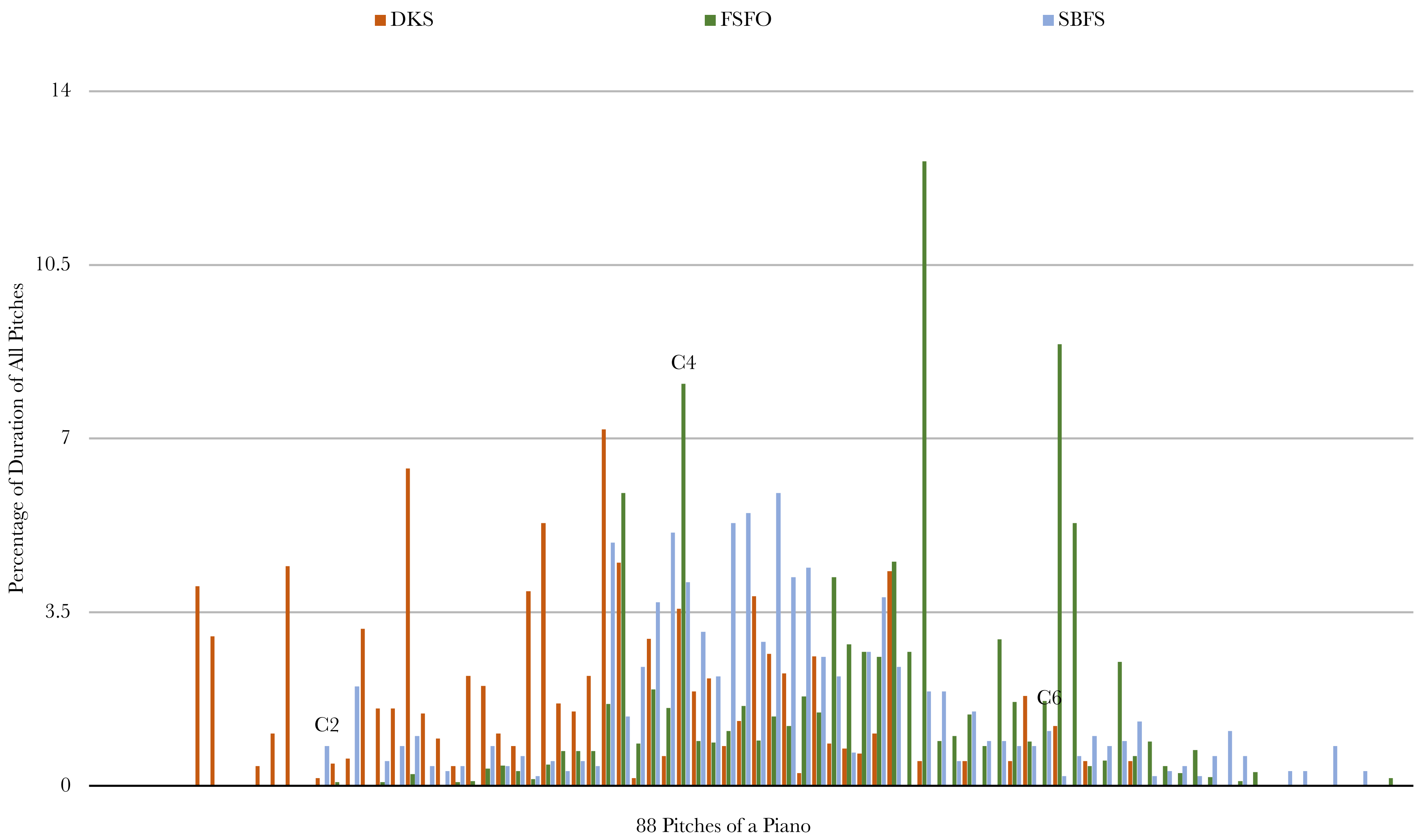


Figure 3-1: Pitches

Pitch Classes

Progressing to the pitch class information, Figure 3-2 shows the data for all three pieces. Although there is variety for each piece, no hierarchical patterns emerge, particularly compared to similar graphs for tonal practice (e.g. Aarden, 2003, p. 82; White, 2013, pp. 81–83). That there are no novelties for any of these pieces makes this point even more clearly: no pitch class is being significantly prioritised above any other. Regarding the IQR, these are: DKS: 7.0; FSFO: 5.4; SBFS: 2.0. The disparities here are noteworthy: clearly DKS is more variable in its distribution of pitch classes than SBFS (a point that even a cursory glance at Figure 3-2 supports). Nonetheless, the lack of novelties is the most significant observation here.

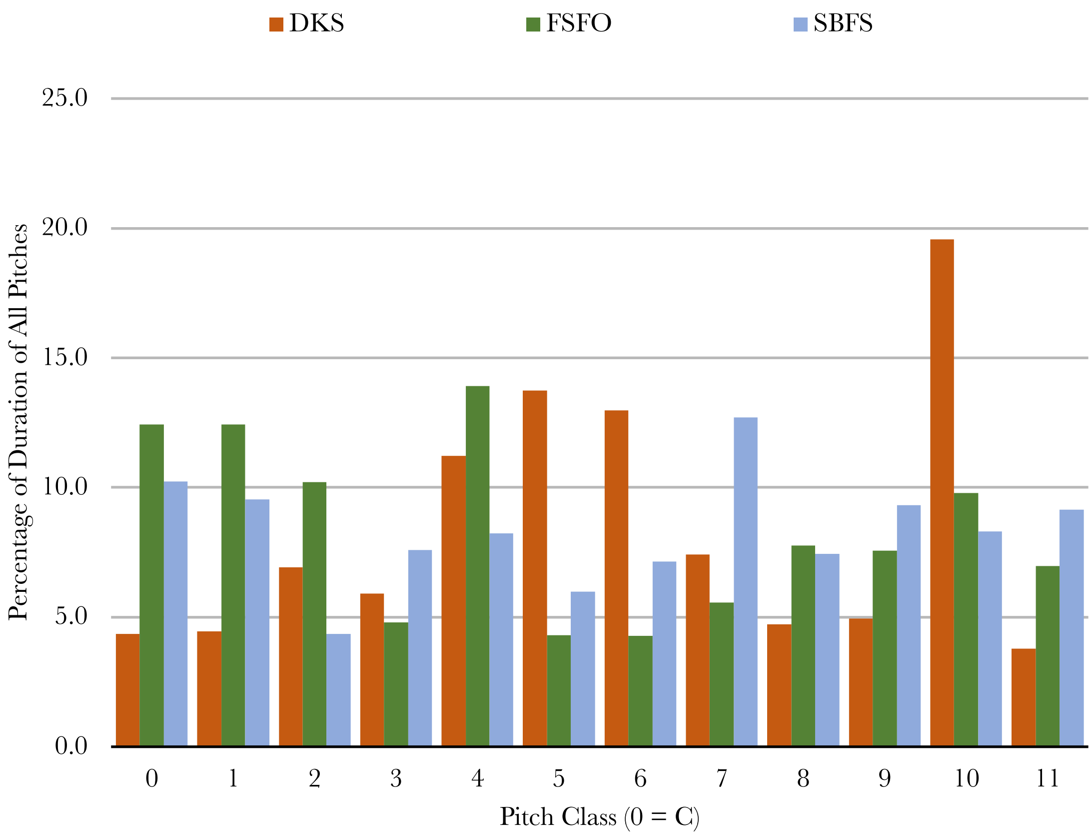


Figure 3-2: Pitch Classes

Density

This data collection also provides information about the densities of the texture, that is, the proportion of these works that consists of sounding simultaneities rather than silences (i.e. the number of notes = 0), and of those simultaneities, the number of notes within each (Figure 3-3). The most apparent feature of this chart is the very high percentages for 0 notes, which indicates the sparsity of Webern's music: almost half of FSFO consists of silence. This observation is reinforced by the number of monads, which is particularly noticeable given the instrumentation of these three pieces: all of these pieces are for multiple instruments, and so some counterpoint might be expected to be typical. Indeed, neither DKS nor SBFS uses any simultaneities larger than a hexad, and 91.9% of FSFO uses simultaneities that are octads or smaller.

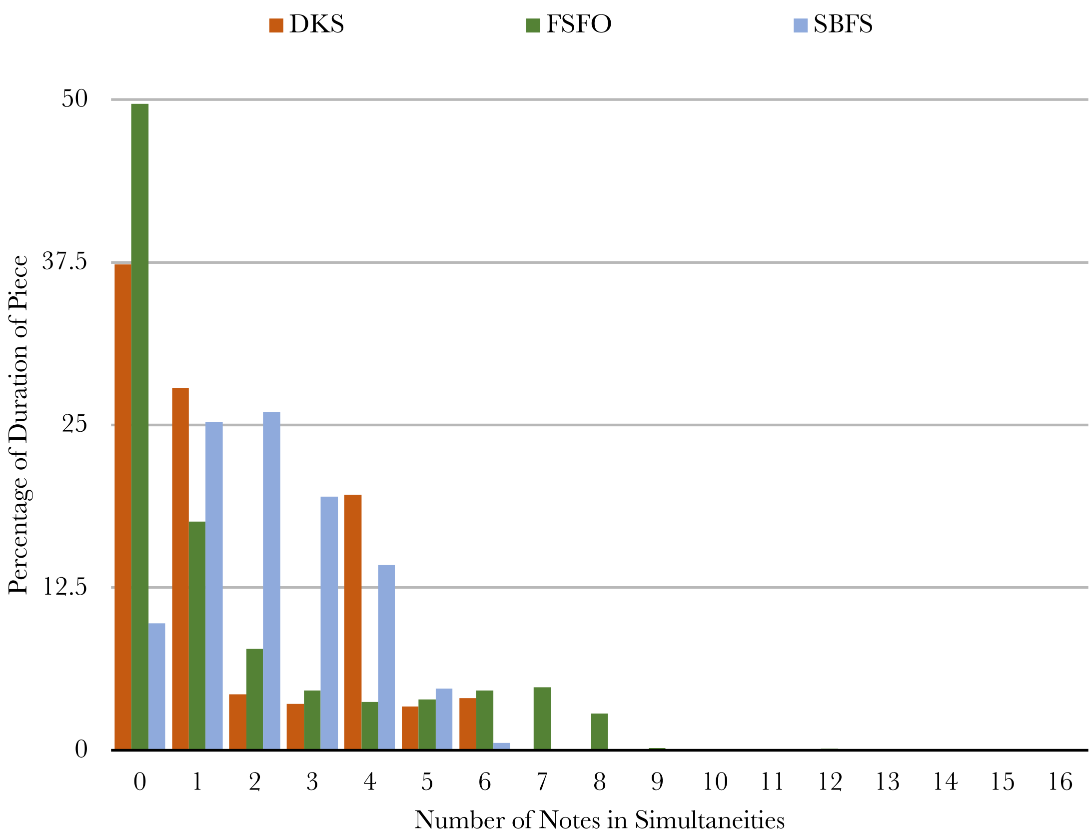


Figure 3-3: Densities

Simultaneities: Totals

Regarding pitch complexes, Figure 3-4 is a histogram of the results: the y-axis presents the number of different simultaneities with a duration within a given span (those ranges on the x-axis). The durations are measured as percentages of the total duration of all the simultaneities in each piece. As for the pitch data above, this is not the same as the total duration of the piece; the decision to consider this data has been made as the interest is in statistical patterns within those simultaneities that Webern chose to use – silences are only of interest in considering textural matters (see Figure 3-3). The positive skew of Figure 3-4 indicates that the vast majority of simultaneities are employed for only a very short period of time, likely with little repetition. Although DKS appears to display a somewhat different pattern to FSFO and SBFS, with a comparatively more even spread across the range 0≤x<2.5 rather than the enormous spike in the initial 0≤x<0.5 range of the latter two pieces, the vast majority of its simultaneities (80 out of 86) are still clustered in this lower end.

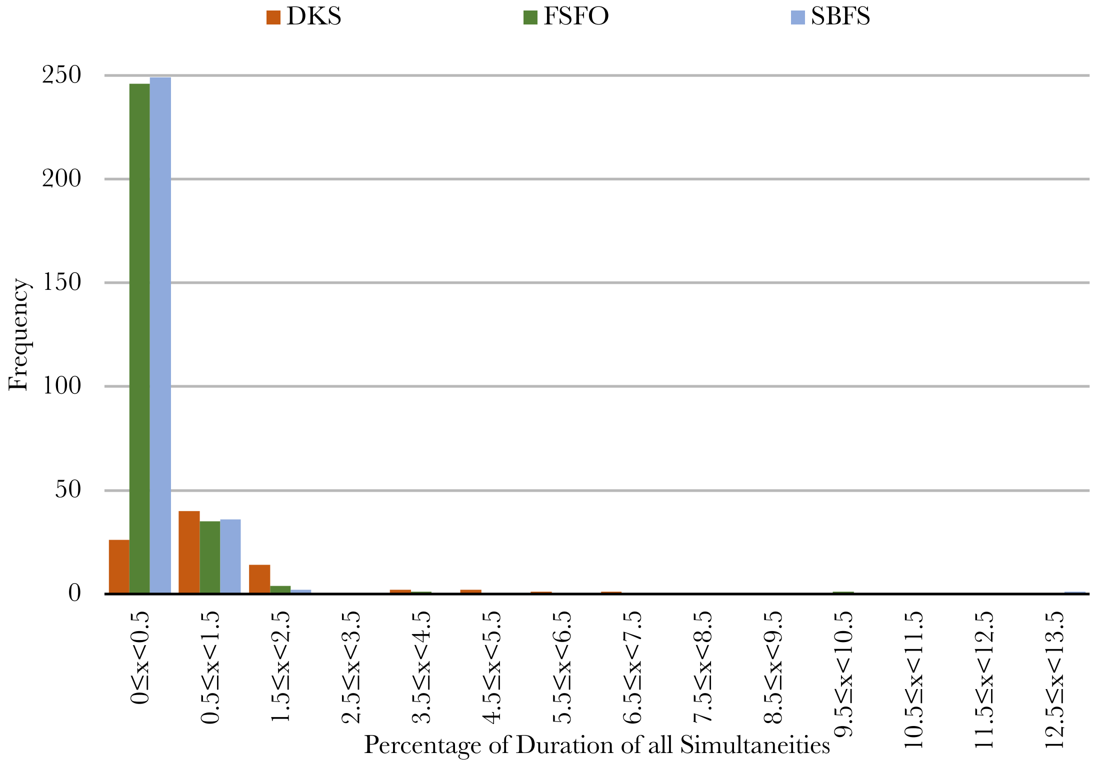


Figure 3-4: Pitch Complex Histogram

Figure 3-5 is a histogram of the distributions of class complexes. The pattern here is even more marked than in Figure 3-4, again with the significant positive skew indicating that almost all of the complexes are used for a very small proportion of the piece. Indeed, in DKS all but one of the complexes fall within the range of 0≤x<7.5, in SBFS, this is true for the range 0≤x<3.5, and in FSFO, 0≤x<2.5. The exception in all three cases is the class complex (C4), corresponding to all monads, which is unsurprising given the density patterns identified above. This falls at: SBFS: 27.5≤x<28.5; FSFO: 34.5≤x<35.5; and DKS: 43.5≤x<44.5. For all three works, (C4) is by far the most-used complex, not only indicating its own importance, but also the overall variety in use of simultaneities.

Reducing the simultaneities one step further, Figure 3-6 presents the distribution of class collections across the three pieces. The same basic shape as before remains clear, with the majority of collections falling within the range 0≤x<2.5 (DKS: 40 out of 44; FSFO: 129 out of 131; SBFS: 82 out of 87); nonetheless, the tail across the graph as a whole is marginally more even, and the initial spike slightly less extreme, than for the complexes. Again, the collection (0) (equivalent to (C4)) is by far the most used, with a proportion of over 30% higher than any other collection for all three pieces.

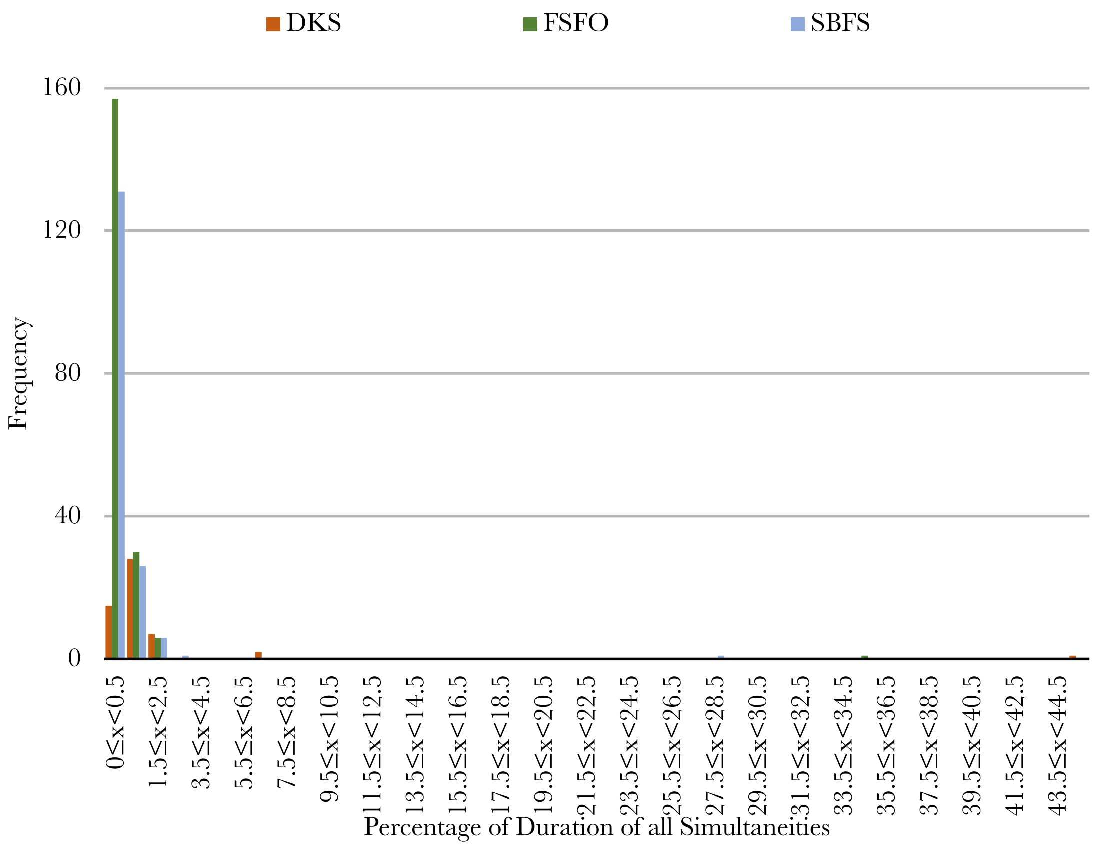


Figure 3-5: Class Complex Histogram

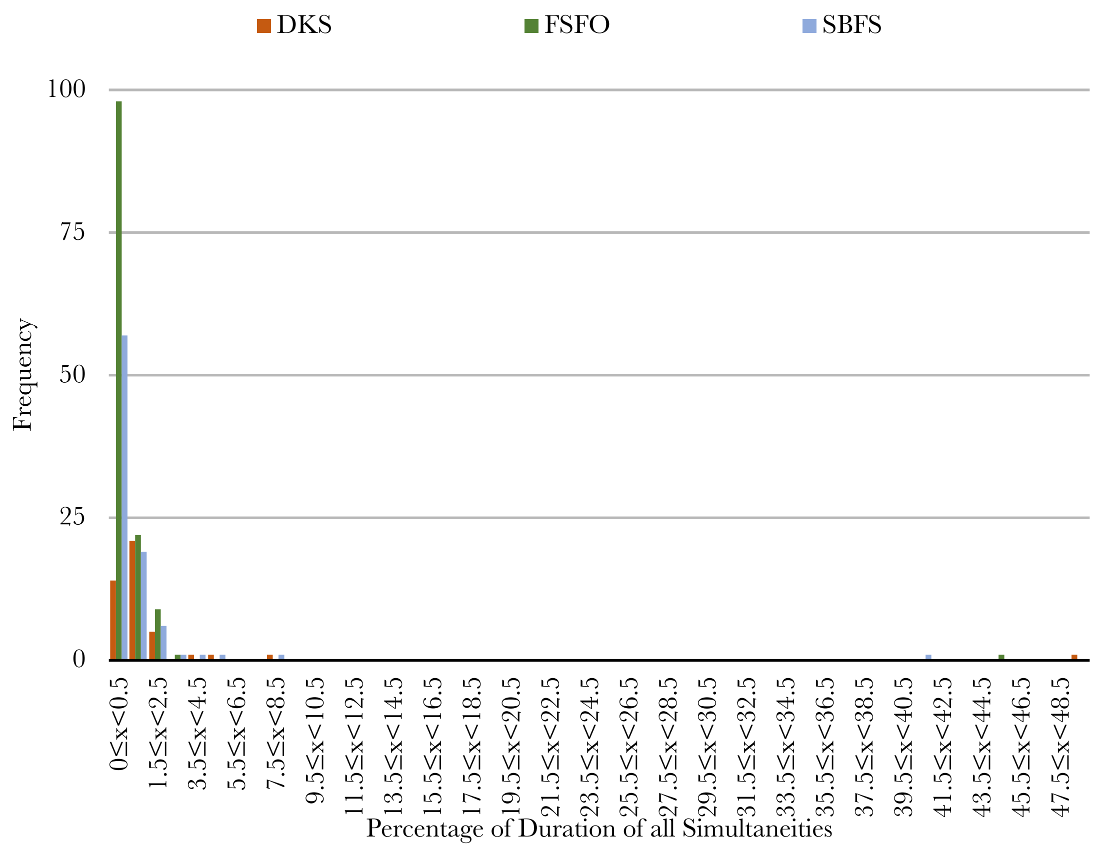


Figure 3-6: Class Collection Histogram

Simultaneities: Novelties

Regarding the pitch complex novelties, DKS has 6, FSFO 23, and SBFS 32. There are no non-monad simultaneities that occur as novelties in any pair of pieces, and (D5) is the only simultaneity to occur as a novelty in all three pieces, explained, in part, by its registral centrality. As to the location of these novelties, the point of interest is whether larger simultaneities (triads or larger) occur scattered across the pieces, or in a concentrated manner. Though the above commentary demonstrates that Webern appears not to have reused significantly identifiable (i.e. large) pitch complexes in novelties between multiple pieces, their disposition within pieces is itself of interest. In the case of DKS, the two larger novelties (the others are all monads) each only occur positioned next to themselves. As for FSFO, the situation is somewhat more complex. 10 out of 23 novelties are classed as 'large', and these 10 essentially comprise five pairs of simultaneities which are closely linked and only appear alternating with the other simultaneity in the pair.

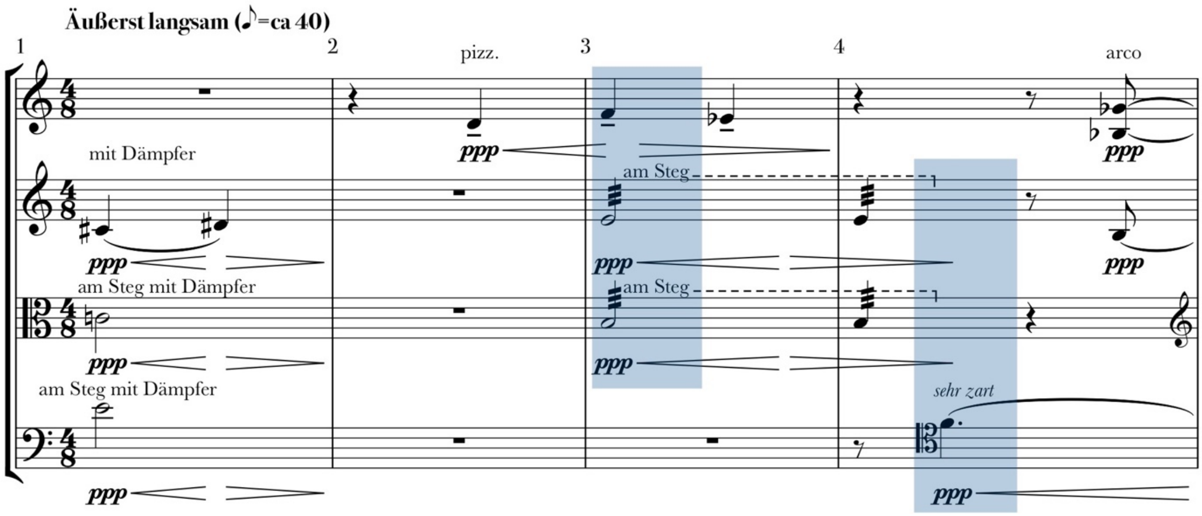
In order to consider these pairs of alternating simultaneities, it is useful to introduce Roig-Francoli here, who defines 'connection' between set classes, and then a more stringent set of criteria for true 'extension'. The fundamentals of connection are thus:

> If the set classes are related by chromatic voice leading, connection results if at least one actual pc in the first set is related to one pc in the second set by common tone or chromatic voice leading. If all pcs between the sets are related by common tone, chromatic, or whole tone voice leading, the sets are maximally connected. (Roig-Francoli, 2001, p. 69)

Meanwhile, 'extension' takes place only if the two sets are totally related by common tone or chromatic voice leading, with the possibility for whole tone voice leading in one voice only (if there is also at least one common tone connection between the sets) (Roig-Francoli, 2001, p. 69). Roig-Francoli goes on to expand his definitions of connectivity to allow for sets of differing cardinality: a larger set is connected to a smaller set if at least one pitch class is related by common tone or chromatic voice leading; a larger set extends, or is extended by, a smaller set only if the smaller set is a subset of the larger set (Roig-Francoli, 2001, p. 70). In Roig-Francoli's work, these ideas of connectivity and extension provide the foundation for his conception of Pitch-Class-Set Extension Regions, into which a piece can be divided. Here, it is useful to employ his definitions as a way of formalising the degree of connection between pairs of simultaneities (and below, sequences of simultaneities). This will be applied not only to strict pitch class sets, as outlined in his theory, but also to the different types of data collected in this thesis, as his ideas are applicable to different types of pitch collection. A consistent aim of this project is to apply the same criteria to different collections of data from the same pieces to yield comparative results.

Using these definitions, the simultaneities in three of the five pairs of alternating novelties in FSFO are maximally connected, and the other two pairs display significant connectivity as they are identical except for one pitch, which differs by a perfect fourth in each case. All three of the maximally-connected pairs thus extend each other, and although the other two do not, their similarity is clear.

In SBFS, three of the nine larger novelties occur only once and four occur side-by-side with themselves, as in DKS. Of the two others, one, (B3 E4 F4), occurs twice, once each in bb.3 & 4 of Movt. V, but with an intervening simultaneity (see Example 3-1). That such a short duration can constitute a novelty is another indication of quite how varied Webern's use of simultaneities is. The final novelty (G3 C#4 F#4) is the only one to occur with significant displacement, as it occurs once in Movt. I, b.10, and four times in Movt. IV, b.5. It is worth acknowledging that these statements are both very brief, however: Webern hardly seems to be drawing attention to this simultaneity, and so the significance of this repetition as particularly marking this out seems limited.



Example 3-1: SBFS, Movt. V, bb.1-4

Considering the novelties for class complexes there are: DKS: 3; FSFO: 18; SBFS: 13. Again, no novelties larger than a monad occur as novelties in any pair of pieces, or in all three works, although (C4) is, as indicated above, the most popular novelty by far in each work. As for longer novelties, in DKS the two are unrelated, and both appear side-by-side with themselves. As with the pitch complexes, in FSFO 10 longer novelties (larger than a dyad) exist in all, which again fit into five pairs, of which three display maximal connectivity, and two significant connectivity. It is also interesting to observe that these two lesser-connected pairs are themselves connected, by the addition of an extra note from one pair to the next, that is, the first pair is (C4 E4 G#4 F#5 G#5 D6 F6) and (C4 E4 C#4 F#5 G#5 D6 F6), and the second is (C4 E-4 E4 G#4 F#5 G#5 D6 F6) and (C4 E-4 E4 C#4 F#5 G#5 D6 F6).[1] As for positioning, all five pairs appear only as alternating with each other, as for the pitch complexes.

In SBFS, there are four larger novelties. Interestingly, however, three of these novelties recur more frequently across the work. (C4 F4 F#4) occurs once each in Movt. V, bb.3 & 4, and once in Movt. VI, b.1; (C4 C#4 D4) occurs once in Movt. I, b.7, once each in Movt. V, bb.6 & 8; and (C4 F#4 B4) occurs once in Movt. I, b.10, once in Movt. III, b.2, and repeatedly in Movt. IV, b.5. The final long novelty, (C4 C#4 G4 G#4), occurs side-by-side with itself.

Turning to class collections, despite the further level of reduction, the situation regarding novelties remains fairly similar. In DKS, there are four novelties, of which two are large, (0 1 2 8) and (0 1 11). The former of these can be understood as strongly linked to the latter: if it is rewritten as (0 1 7 11) then it can simply be seen as comprising the second collection with the addition of one note; nonetheless, they do not occur in close proximity, and neither collection is reused across the works.

In FSFO, the usual popularity of dyads prevails: 9 out of 13 collections are either monadic or dyadic. The remaining four comprise two pairs of collections, each differing by the addition of one note: (0 2 4 5 6 8) & (0 1 2 4 5 6 8), and (0 1 4 5) & (0 1 3 4 5). Each of these pairs appears with their collections alternating with each other, so again lacking widespread disposition across the pieces, and as the smaller sets are all subsets of the larger sets, they produce extension.

Regarding SBFS, 5 out of 11 novelties are large. Four out of five of these appeared as class complex novelties. Regarding disposition, these are, again, more widely spread. (0 1 8 9) occurs in Movt. II, bb.6 & 7, and Movt. III, b.9; (0 6 11) in Movt. I, bb.6 & 10, Movt. IV, bb.1 & 5, and Movt. V, bb.3 & 4; (0 5 6) in Movt. I, b.10, Movt. II, b.1, Movt. III, bb.2 & 4, Movt. IV, bb.1 & 5, Movt. VI bb.1 & 2; and (0 1 2) in Movt. I, b.7 and Movt. III, bb.5 & 6. That these are much more spread out than the class complex novelties indicates that whilst these collections are fairly wide-ranging, they often appear in varied voicings; only (0 1 7 8) appears just once.

[1] In music21, a minus-sign is used to indicate a flat, so E-4 = Eb4.

Simultaneities: Subsets and Supersets

A further way of considering simultaneities that are of particular interest is by looking at subset and superset relations. This basic concept is outlined and employed by Forte (1973, pp. 24–46), but in this project it will be used in a somewhat different manner. Here, small simultaneities that are used in the piece both on their own and as subsets of larger entities have been identified. Precisely, 'small' defines those simultaneities with three or four elements: monads or dyads are so basic as to be of minimal significance. As the interest is in these smaller 'types' of entity, they are considered only in terms of class complexes and class collections. This indicates small 'basic types', which form a foundational part of the larger entities in the pieces.

Figure 3-7 is a histogram that shows the frequency of subsets with a given number of resulting supersets in each piece, for class complexes. As a clarifying example, DKS has three simultaneities, (C4 C#5 D6), (C4 F4 F#5), and (C4 E4 B4 D5), which each make up part of exactly one superset that is also used in DKS: respectively, (C4 G4 C#5 D6), (C4 F4 D5 F#5), and (C4 E4 B4 D5 G#6). The figures for lower numbers of related simultaneities are of less interest here: of greater significance are those entities which form part of a high number of supersets, which are really only a feature of SBFS and, particularly, FSFO.

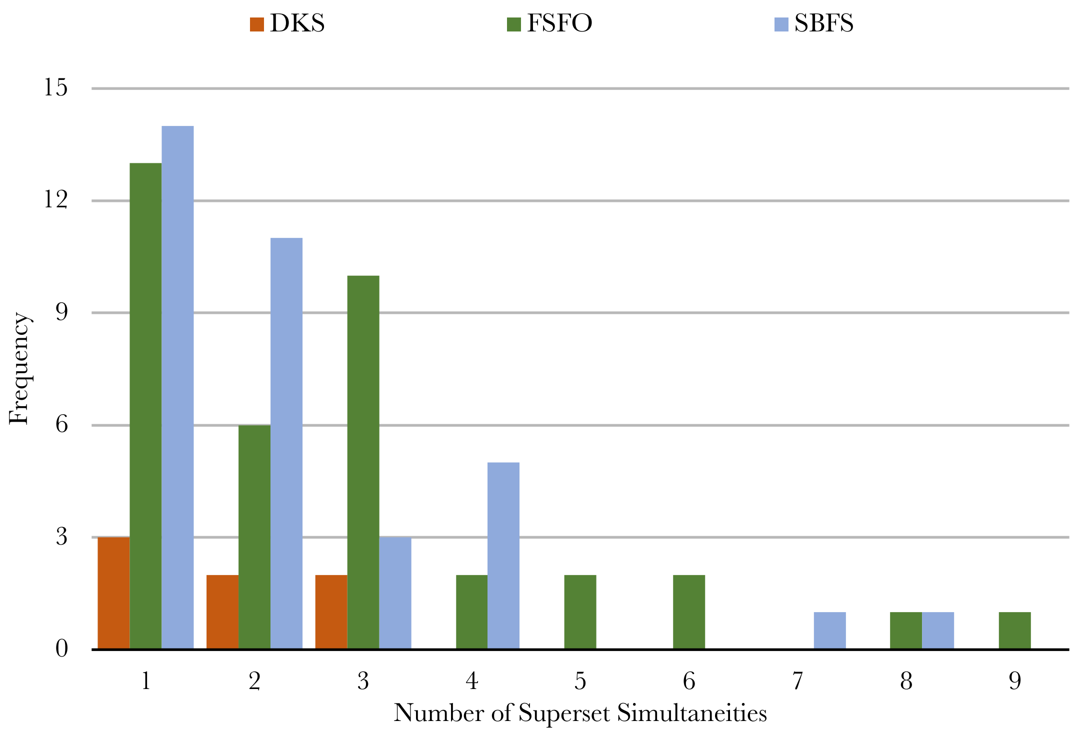


Figure 3-7: Class Complex Superset Histogram

As for class collections, Figure 3-8 is a histogram displaying the frequency of different subset simultaneities by the number of supersets to which they are related. As before, DKS has very few; however, here FSFO and SBFS are far more distinct, with FSFO hugely outstripping SBFS. Likewise, the total durations of the resulting supersets in FSFO are far greater than SBFS: the longest duration for the supersets of any single subset in SBFS is 8s; FSFO has 15 subsets with related supersets that have a long duration than this.

Particularly significant in FSFO are those subsets with both a high number of related supersets *and* a long total duration for those supersets: this indicates particularly high significance. Although the latter of these criteria may appear to be a logical implication of the former, this is not the case: the subset (0 1 9) with the second-highest number of supersets, 22, has a total resulting duration of only 9s. Conversely, there are seven subsets with over 10 supersets with a cumulative duration over 15s: (0 1 4), (0 1 5), (0 4 6), (0 6 8), (0 2 4), (0 1 4 5), and (0 5 8). Probably unsurprisingly, but notably nonetheless, most of these collections avoid any tonal implications. The primary exception is (0 5 8), a second-inversion minor triad. Considering the supersets in which (0 5 8) occurs, however, indicates that the larger collections have no bearing on tonal harmony: the collections always have five or more notes, and always employ semitonal clusters. The other possible group of collections here that have more traditional implications are whole-tone, as in (0 2 4), (0 4 6), and (0 6 8). Again, however, contextualising these shows Webern frequently surrounding these subsets with semitones, avoiding any possible whole-tone suggestions.

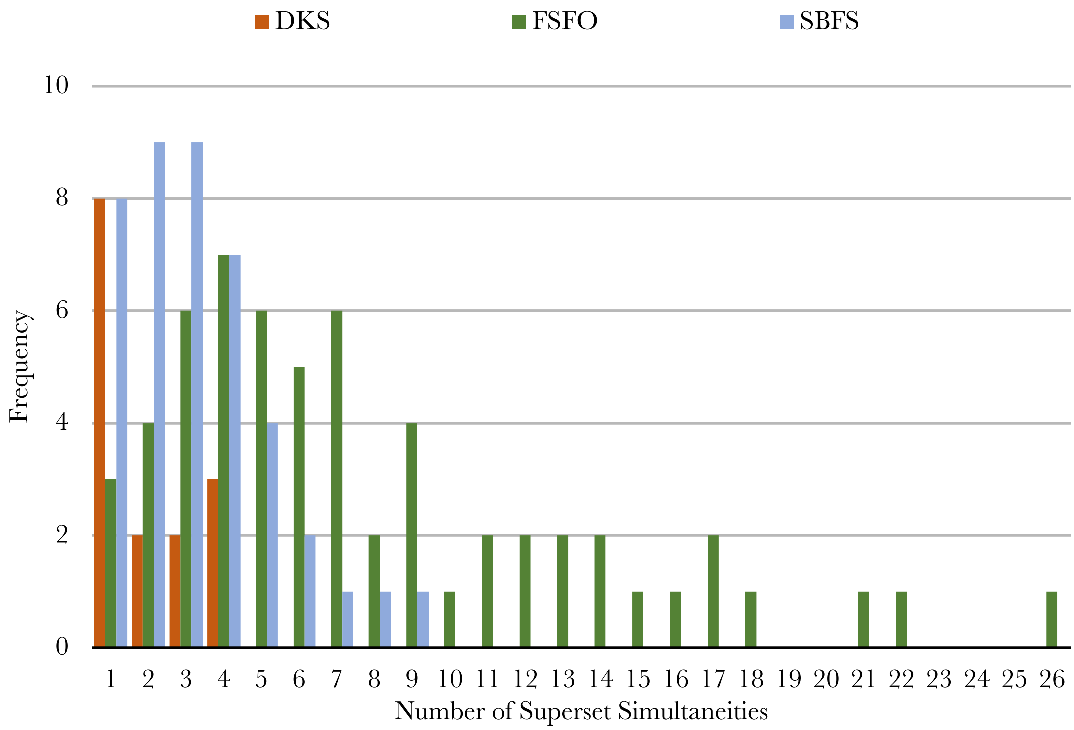


Figure 3-8: Class Collection Superset Histogram

Simultaneities: Mutual Simultaneities

The final area of interest is those simultaneities that occur in multiple works. The number of mutual pitch complexes is as follows: DKS & FSFO: 5; DKS & SBFS: 5; FSFO & SBFS: 13; all three: 12. These are potentially less significant than those mutual simultaneities discussed above, as they are not necessarily novelties, and so may not be significant even within individual pieces. Their significance declines further in the light of the size of these simultaneities: all but four are monads, and those four are all dyads (one each in DKS & FSFO and DKS & SBFS, and two in FSFO & SBFS). This indicates that there are no large, and therefore easily characteristic or recognisable, pitch-complex simultaneities that Webern is reusing at all between these pieces.

Considering total mutual class complexes, the figures are as follows: DKS & FSFO: 2; DKS & SBFS: 5; FSFO & SBFS: 16; all three: 9. Again, of these 32 only nine are not monadic or dyadic complexes. These nine are all triads, and none occur in all three pieces. Further, none of these appears as novelties in any of the three pieces; thus, although there are larger complexes that Webern is reusing between pieces, again he is not prioritising them through proportional significance in any of the works.

As for class collections, the figures are unsurprisingly much higher than before: DKS & FSFO: 10; DKS & SBFS: 4; FSFO & SBFS: 22; all three: 12. All possible dyads occur, as does the monad. Regarding those larger collections that appear in all three pieces, they are all triads, and all include either (0 1) or (0 11). More broadly, many of the novelties that occurred in each of the pieces recur in other pieces, even if not as novelties, such as from DKS (0 1 11) and (0 1 2 8), from FSFO (0 1 3 4 5), and from SBFS (0 1 8 9), (0 5 6), (0 1 2), and (0 6 11).

Sequences: Totals

Turning to sequences of pitch complexes, with both two and three complexes, the overarching trend is of heterogeneity. Table 3-1 provides the number of different sequences in each work: the figures are very high, particularly given the tiny duration of these works. The other major overall observation is that the vast majority of sequences are stated only once. Indeed, for all categories the IQR is 0, as the number of statements of a sequence at both the first and the third quartiles is 0.

The situation with class complex sequences is very similar to that of pitch complex sequences. As Table 3-2 shows, there are very high numbers of different sequences, and the IQR remains 0, with both the first and the third quartiles occurring at one statement.

Piece	Sequence Length	Sequence Count
DKS	2	102
	3	108
FSFO	2	375
	3	425
SBFS	2	714
	3	746

Table 3-1: Pitch Complex Sequences

Piece	Sequence Length	Sequence Count
DKS	2	76
	3	91
FSFO	2	299
	3	355
SBFS	2	274
	3	331

Table 3-2: Class Complex Sequences

As before, the relationship between class collections and class complexes is very similar. Whilst the total numbers of sequences are inevitably slightly smaller (Table 3-3), there is again an IQR of 0.

Piece	Sequence Length	Sequence Count
DKS	2	72
	3	89
FSFO	2	268
	3	341
SBFS	2	254
	3	315

Table 3-3: Class Collection Sequences

Sequences: Novelties

A further effect of the heterogeneity is that novelties, calculated as before, comprise any sequence that is used more than once. Figure 3-9 provides a histogram of these novelties for pitch complexes, clearly demonstrating how rare they are. That FSFO has so many novelties (48 two-complex and 46 three-complex) is in part due to some of the repetitive textures in the work (e.g. Example 3-2). Whether this is a drawback of the methodological approach is questionable: arguably these sorts of harmonically static textures do not lend significance to collection-sequences in the way that repeated statements of a sequence across the course of a work would; nonetheless, repeating these sequences certainly increases their importance, as does keeping the texture dynamic in this way. Considering the novelties more broadly, there are no sequences which appear as novelties in any pair of the pieces, or in all three pieces. As for DKS, there is only one novelty at all: (E-2) (C4). This is a remarkable situation – that across 210 different sequences, there is only one repetition (this sequence is used twice), and each simultaneity in this sequence is only one note.

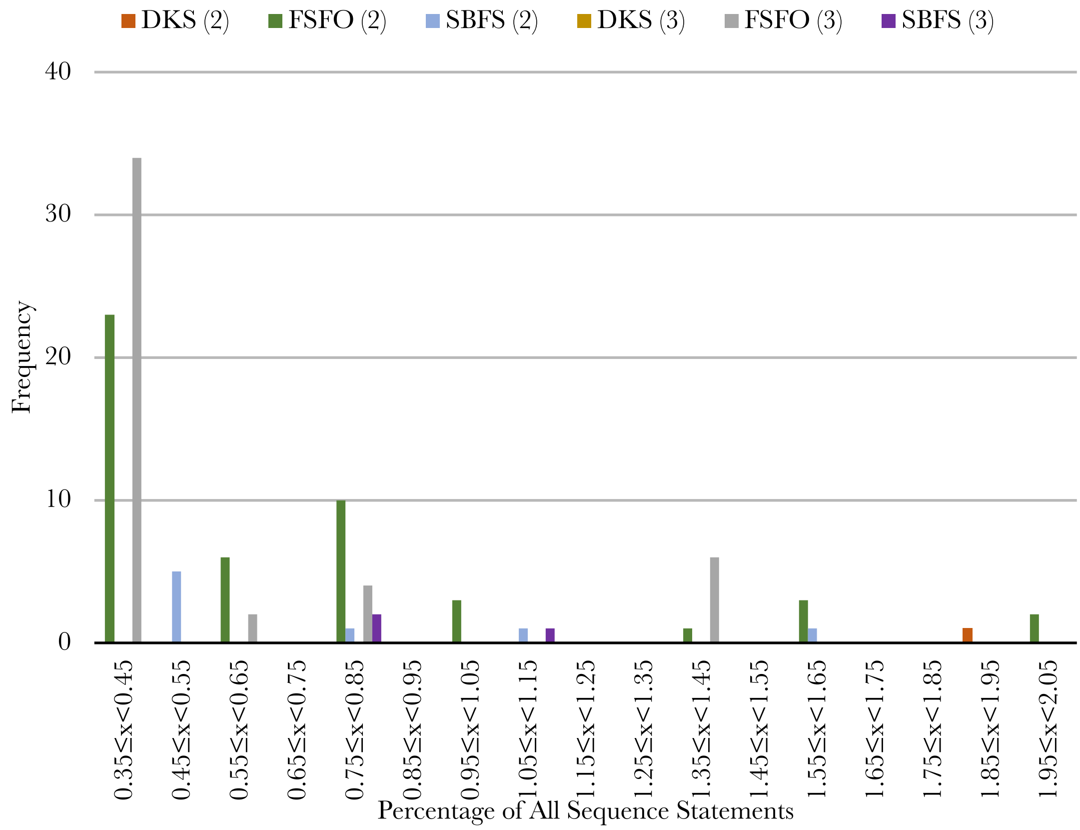


Figure 3-9: Pitch Complex Sequence Novelty Histogram[2]

---

[2] In the sequence novelty histograms, the data are categorised by length of sequence within each piece, indicated by the number in parentheses following the piece-name.

Example 3-2: FSFO, Movt. III, bb.8-11

Anton Webern, '5 Stücke für Orchester, Op. 10' © 1923, 1951 by Universal Edition A.G., Wien/UE5967

Returning to FSFO, in processing this greater number of novelties, one meaningful criterion is the degree of similarity between the complexes in the sequence. The greater the disparity between the complexes, the more significant, as these are more likely to be identifiable and characteristic. In this light, then, it is noteworthy that 37 out of 48 of the two-complex sequences, and 74 out of 92 of the sub-pairs within the three-complex sequences (i.e. (*ab*) and (*bc*) from a sequence (*abc*)), have no or a one note difference between the complexes. Of these, however, only 8 two-complex sequences and 13 sub-pairs have maximal connectivity (in total, 15%). This indicates that those sequences that Webern does choose to repeat are hardly dramatic shifts of pitch content; rather, most of the time he is either repeating a complex or changing one note. These changes of note are often greater than a semitone or whole-tone (indicated by the small maximal connectivity figures), but the changes remain limited.

SBFS demonstrates similar trends. Here, all of the two-complex sequences, and the sub-pairs within the three-complex sequences, have no or one note differences between the complexes. Again, maximal connectivity is comparatively rare, occurring in one two-complex sequence, and both sub-pairs of a single three-complex sequence (in total, 21%).

For class complexes (Figure 3-10) the classification of novelties is again any statement that is used more than once. There are some here mutual novelties here. Unsurprisingly, (C4 C4) and (C4 C4 C4) occur as novelties in all three pieces and, with the exception of three-complex sequences for FSFO, are always the most common sequence. As for pairs of pieces: (C4 C#5) (C4) occurs in DKS & SBFS; (C4 E4) (C4) in DKS & FSFO; and (C4) (C4 F4), (C4) (C4 D4), (C4 F4) (C4), (C4 D6) (C4), (C4) (C4) (C4 F4), (C4) (C4) (C4 D4) in FSFO & SBFS. It is notable that in every case for two-complex sequences one of the complexes is (C4), and for three-complex sequences two of the complexes are (C4), and the other complex is a dyad. Given the density patterns identified above, which indicate the prevalence of (C4) as by far the most popular class complex, this is largely to be expected. The implication of this is that rather than reusing patterns of class complexes with particularly notable identities, the only complexes to be regularly reused are dyads, and the likelihood is that they will be positioned next to monads due to the sparse overall texture.

As for similarities between the complexes within these sequences, for DKS every sequence or sub-pair within three-complex sequences includes complexes with no or a one note difference, for FSFO 101 out of 123 do, and for SBFS 63 out of 69. Maximum connectivity is somewhat higher this time: respectively, 5, 30, and 20, or 50%, 24%, and 29%.

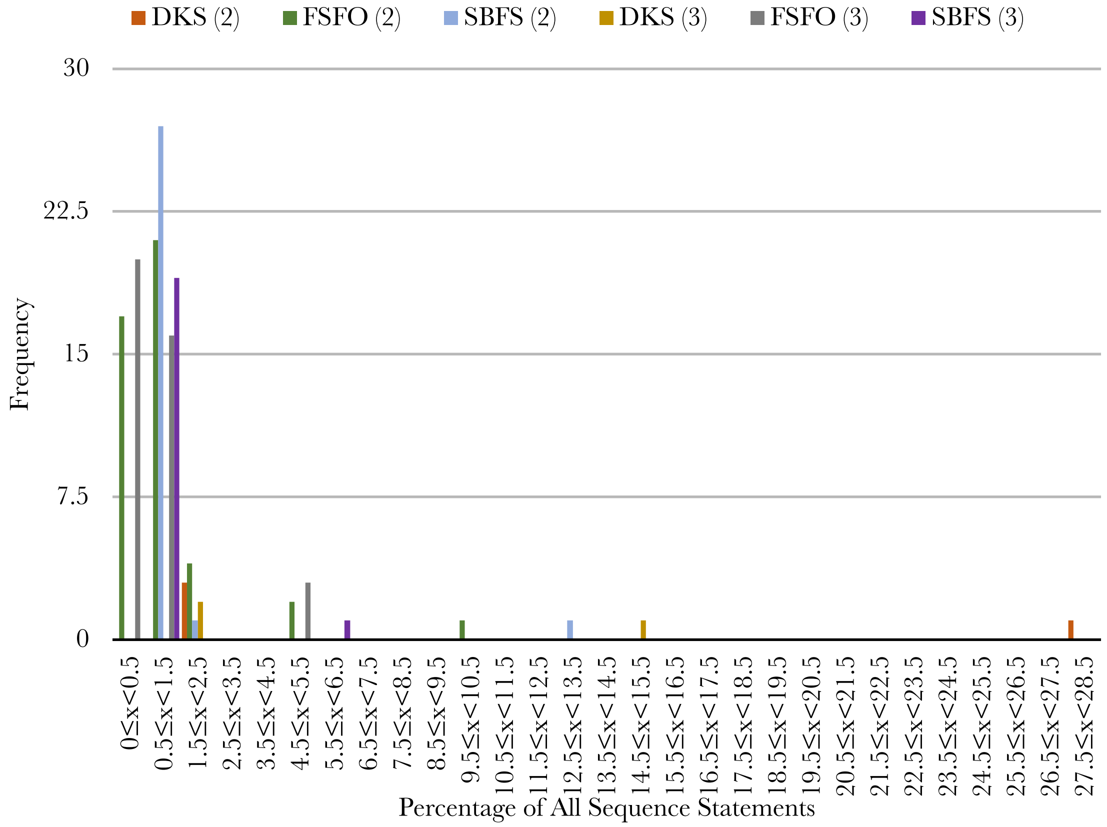


Figure 3-10: Class Complex Sequence Novelty Histogram

For class collections the pattern of novelties as shown in Figure 3-11 is almost identical to that of Figure 3-10. Regarding mutual novelties, (0) (0), (0) (0 5), (0) (0 6), and (0) (0) (0 6) occur in all three pieces. (0) (0 1), and (0 1) (0) occur in DKS & SBFS; (0) (0) (0), (0) (0) (0 5), and (0 4) (0) in DKS & FSFO; and (0) (0) (0 2), (0) (0 2), (0 8) (0), (0 5) (0), (0 2) (0) (0), (0 2) (0), and (0 11) (0) in FSFO & SBFS. As before, monads are very common, and all other collections are dyadic.

Considering similarities between the complexes in these novelties, for DKS 17 out of 18 pairs of collections in novelties have either no change or else a one note difference, for FSFO 112 out of 136 do, and for SBFS 72 out of 117 do. Maximum connectivity is lower, however: respectively, 8, 14, and 27, or 44%, 10%, and 23%.

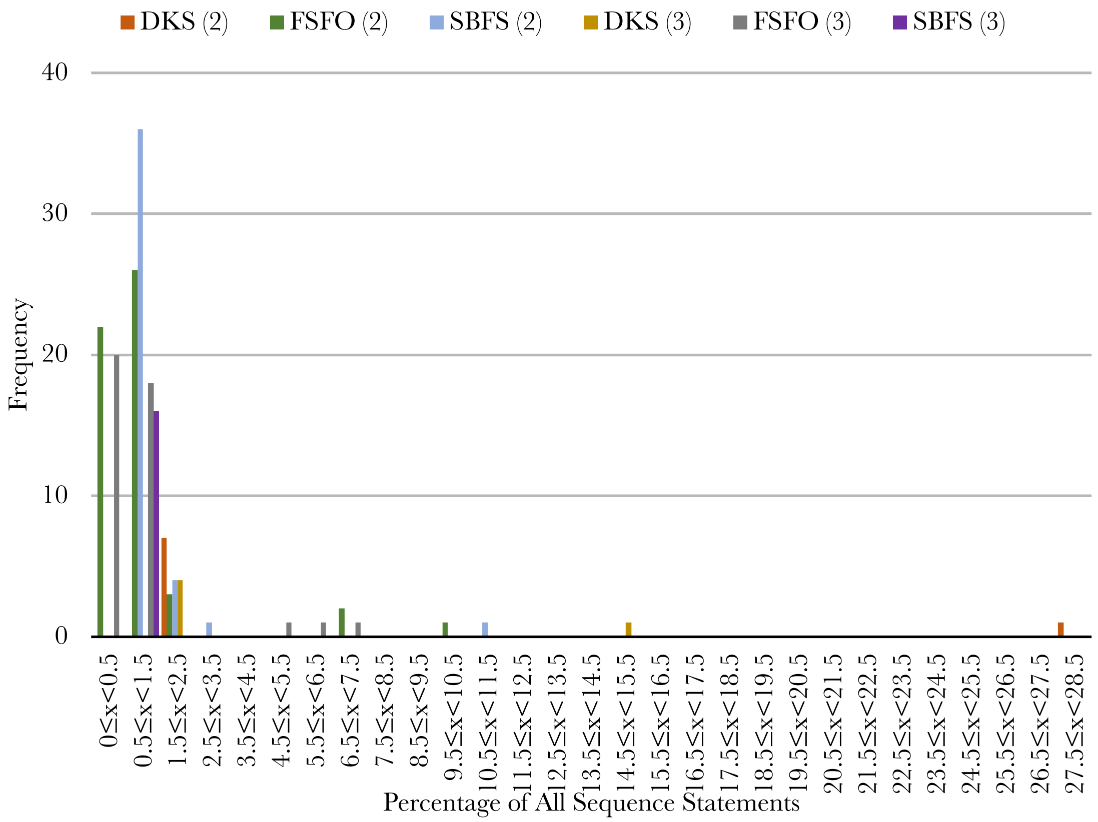


Figure 3-11: Class Collection Sequence Novelty Histogram

Sequences: Mutual Sequences

Considering mutual pitch complex sequences, there are none which recur as novelties between pieces, and only one which recurs between two pieces. This is (C5) (D4), which occurs in FSFO & SBFS. This is further testament to Webern's ingenuity: it is remarkable that only one sequence is reused across these pieces, and especially so, given that very sequence is only a succession of two individual notes.

Mutual sequences are unsurprisingly much more common for class complexes than with pitch complexes, as Table 3-4 demonstrates. Nonetheless, similar patterns to mutual novelties occur. Only one two-complex sequence does not include (C4) as one of the complexes, (C4 F5) (C4 E4), and only one three-complex sequence does not have (C4) as two out of the three complexes, though it has (C4) as one of the complexes, (C4) (C4 F5) (C4 E4). Dyads and triads remain very common too: 18 of the two-complex sequences have dyads as at least one of their complexes, with the remainder including triads, and in the three-complex sequences every non-(C4) complex is a dyad.

Pieces	Two-Complex Sequences	Three-Complex Sequences
DKS & FSFO	5	1
DKS & SBFS	6	4
FSFO & SBFS	7	5
All Three	4	3

Table 3-4: Number of Mutual Class Complex Sequences

As for class collection mutual sequences, this is lower only for DKS & FSFO two-collection sequences. 30 out of 42 two-collection sequences contain (0), 17 out of 21 three-collection sequences contain (0) (0), and every other includes (0). Meanwhile, 26 sequences include a dyadic complex, 12 include a triad as a complex, four sequences include a tetrad, and one sequence a pentad.

Pieces	Two-Collection Sequences	Three-Collection Sequences
DKS & FSFO	3	1
DKS & SBFS	8	6
FSFO & SBFS	23	10
All Three	8	4

Table 3-5: Number of Mutual Class Collection Sequences

Chapter 4: Discussion

Introduction

This chapter will synthesise the results presented above, drawing broader conclusions from the body of data to explore some implications about the nature of harmony in the three pieces under consideration. It will therefore attempt to answer the research questions identified at the outset of this thesis, in particular by looking at trends in the different types of data accumulated from the pieces in aggregate, as well as by applying particular types of statistical enquiry to the data themselves.

These conclusions will also be compared to Jackson's work (1970). Although his corpus was small, dealing only with individual movements or isolated passages, and thus of limited relevance, it serves a useful reference point as the most similar corpus study considering similar repertoire. Following these comparative comments, the discussion will consider SBFS as a case study, exploring in context some of the features highlighted by this study. One of the key strengths of a data-based approach like this is not only that the data can reveal interesting trends and phenomena on their own, but that they can set a baseline for identifying unusual features which require more detailed, contextual consideration: what the data-journalist Nate Silver terms the 'last mile problem' (Klein & Silver, 2018, 00:41:30). As such, the case study seeks to show how data-based enquiry and more conventional contextual analysis might intersect, especially with regard to the analytical work that has already been carried out on SBFS.

Pitches & Pitch Classes

Regarding pitches, there are few comments to make. Clearly whilst the overall distribution is similar between pieces, the weightings of the works differ. Figure 4-1 shows the distributions for each piece by octave, confirming the trends identified above: DKS is notably lower, SBFS notably higher, and FSFO somewhere in the middle. Unsurprisingly, novelties fall centrally, close to the median pitch; indeed, as would be expected, on the whole the pitches themselves cluster in the middle: very high or low pitches tend to be exceptional, establishing a clear registral hierarchy.

Considering pitch classes, as discussed above no real hierarchies emerge. In DKS & SBFS those novelties identified at the pitch level do not replicate in terms of pitch classes. As for FSFO, the distribution of pitch classes is more closely related to the pitch novelties. Table 4-1 shows the seven pitch novelties in FSFO alongside the seven most popular pitch classes, with some clear correlations, particularly at the upper end. Nonetheless, there are no pitch class novelties in FSFO, and so Webern maintains his strategy of avoiding prioritising any single pitch class, and thus any possibility of tonicisation. As Webern put it himself, 'All twelve notes have equal rights' (Webern, 1963, p. 52).

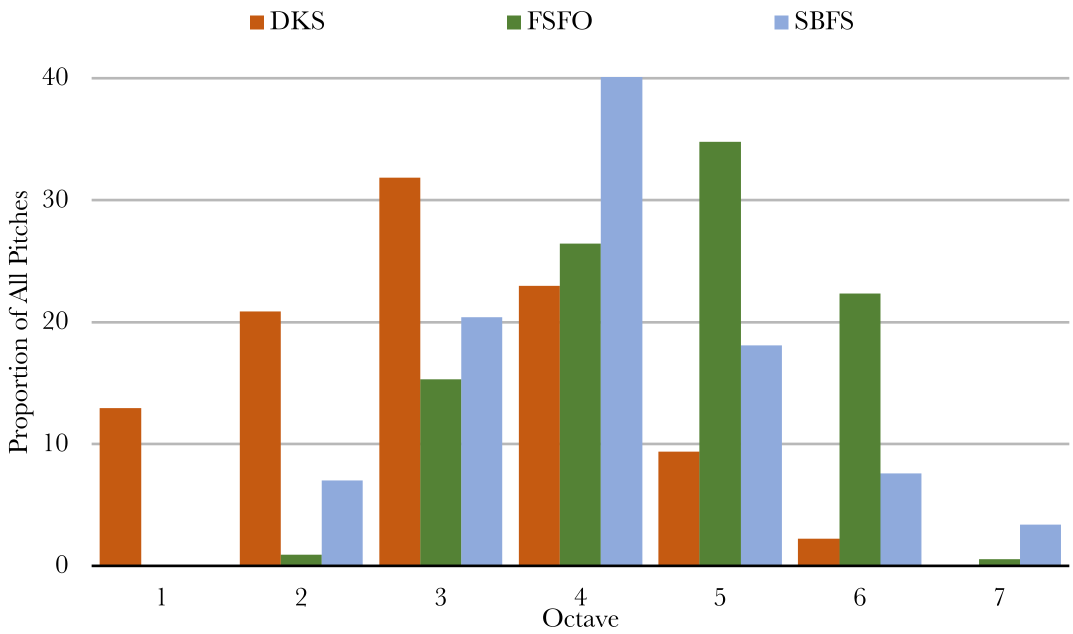


Figure 4-1: Pitch Distributions

Rank	Pitch	Pitch Class
1	E5	E
2	C#6	C#/C
3	C4	C#/C
4	G#3	D
5	D6	A#
6	D5	G#
7	A#4	A

Table 4-1: FSFO Pitch Novelties & Ranked Pitch Classes

Density

In considering the densities of these works, Jackson's study is particularly useful. He proposes that late tonal music had a greater focus on tetradic harmonies, whilst music of the aesthetic under consideration in this project was characterised by a combination of simpler textures or more complex chords. He argues that Webern tended to use a much higher proportion of rests than other composers and that monads were more prevalent in his work, similarly to Schoenberg's (Jackson, 1970, pp. 134–136). Before directly comparing his work with the present study, it is important to distinguish them methodologically: in particular, whilst this project considers density from the number of *pitches* employed in a given simultaneity, Jackson considers the number of *pitch classes*. For considering a textural matter like this pitches is a better metric than pitch classes as it takes account of the actual density of the simultaneities, but the discrepancy is nonetheless small, and comparison is still viable.

The results of this work support Jackson's comments about the situation in later pieces (this project does not have any data from earlier works which can be compared). There are some caveats, however. Whilst the broad observation about the prevalence of silences applies to FSFO (37%) & DKS (50%), it is less apparent in SBFS (10%). This perhaps indicates that Webern's textural style is somewhat more heterogeneous than expected, but also that there may be a something of an increasing chronological trend here. Likewise, whilst Jackson's argument about monads is reflected above, there is less evidence for the more complex (larger than tetrads) harmony that he views as characteristic, with, respectively, 5% of SBFS, 16% of FSFO, and 7% of DKS falling into this category. It is notable in the context of this to consider the spike in tetrads in DKS. This perhaps suggests that Webern utilised the piano as more of an harmonic instrument than a contrapuntal one, in something of a traditional manner; this is exactly the sort of finding that could be augmented by Silver's 'last-mile' analysis. Overall, the finding of this sparsity supports the popular image of his music as pointillistic and delicate – recall Boulez's description of 'the presence of silences in unaccustomed amplitude' (Boulez, 1968, p. 384). This demonstrates the value of carrying out this sort of empirical analysis, as this is no longer a vague assertion, but a quantifiable fact.

Simultaneities

Examining the simultaneities themselves reveals a number of different trends. The first matter to consider is the overall quantity of different simultaneities, and the change in these depending on the level of harmonic reduction (Table 4-2). As is evident, whilst FSFO & SBFS have approximately identical numbers of pitch complexes, this is unsurprising given their identical durations, FSFO has almost twice as many class collections as SBFS. Therefore, whilst SBFS has fewer collections, essentially a proxy for 'types' of simultaneity, they are much more dispersed in terms of pitch level.

Again unsurprisingly, DKS has far fewer simultaneities in each category. Nonetheless, the percentage change between class complex and class collection is significantly smaller than any other change, indicating that in this piece pitch class collections tend not to be repeated with different voicings.

Category	DKS	FSFO	SBFS
Pitch Complex	86	287	288
% Change	-38.4	-32.4	-42.4
Class Complex	53	194	166
% Change	-17.0	-32.5	-47.6
Class Collection	44	131	87

Table 4-2: Total Simultaneities

With regard to mutual simultaneities, Table 4-3 shows the total counts for each pair. FSFO & SBFS always share the greatest number of simultaneities. However, the second highest is always all three, indicating that, to some degree, Webern reuses simultaneities across pieces, although the absolute figures are fairly low. Many of these are, however, small entities, making repetition almost inevitable.

Category	DKS & FSFO	DKS & SBFS	FSFO & SBFS	All Three
Pitch Complex	5	5	13	12
% Change	-60.0	0.0	23.1	-25.0
Class Complex	2	5	16	9
% Change	400.0	-20.0	37.5	33.3
Class Collection	10	4	22	12

Table 4-3: Mutual Simultaneities

A further useful metric concerns the proportion of unique simultaneities in each piece (Table 4-4). Clearly in almost all cases the majority of simultaneities in each piece is unique to it, with the only exception the class collections of DKS. Indeed, the figures for DKS are always smaller than the other two pieces. This would appear to reinforce the hypothesis above of a vocabulary of basic harmonic 'types'. Given the lower number of total simultaneities in DKS, compared to the other two works, if there is a foundational set of simultaneities common to all three works, this will make up a bigger proportion of simultaneities in DKS than in the other two works.

Considering the nature of this foundational group of simultaneities, it is notable that there are no simultaneities larger than a monad that occur as novelties in all three works, at any level of harmonic reduction. This implies that within the hypothesised vocabulary of types, the only simultaneities that are particularly significant within the works themselves tend to be monadic: elementary units.

Piece	Simultaneity Type	% Unique Simultaneities
DKS	Pitch Complex	74.4
	Class Complex	69.8
	Class Collection	40.9
FSFO	Pitch Complex	89.5
	Class Complex	86.1
	Class Collection	66.4
SBFS	Pitch Complex	89.6
	Class Complex	81.9
	Class Collection	56.3

Table 4-4: Percentages of Unique Simultaneities

The percentages of simultaneities that are novelties in each work are displayed in Table 4-5. The first observation to make is how stable the proportions are at different levels of reduction: changes are all less than 5%. To a degree this is explained by the categorisation of novelties, which is governed not by an absolute cut-off, but rather by the distribution of the data themselves. DKS & FSFO tend to be closer, with SBFS as something of an outlier, typically with higher proportions, indicating a greater number of entities which are individually used many times. The significance of this trend is reinforced by considering the location of the novelties, in particular the larger ones. In SBFS these tend to be far more frequently spread out across the piece, rather than clustered in specific locales which are defined by static PCSE, as in FSFO. Indeed, as discussed above the majority of novelties in FSFO fall into alternating pairs, which tend to display PCSE, thus creating passages of quasi-static harmony.

Table 4-6 presents the counts of subsets and supersets in each piece: the counts for subsets indicate the number of triads and tetrads that are used in each piece and recur in at least one larger superset; the counts for supersets show how many supersets in each piece are related to such a subset. Whilst the general increase is expected with increasing harmonic reduction, great variation can be observed. SBFS is something of an outlier in the relationship between the number of subsets and the number of supersets: those subsets that are used make up greater numbers of resulting supersets than in the other two works, particularly in terms of class collections. To a degree this is expected: recalling Table 4-2, the simultaneities in SBFS undergo far more extreme concentration at different levels of harmonic reduction than either DKS or FSFO; nonetheless, that does not make this outcome inevitable. Instead, this indicates a body of subsets that far more frequently recur as parts of larger harmonic entities, perhaps suggesting a more homogeneous set of simultaneity types.

Category	DKS	FSFO	SBFS
Pitch Complex	7.0	8.0	11.1
% Change	-1.3	1.3	-3.3
Class Complex	5.7	9.3	7.8
% Change	3.4	0.6	4.8
Class Collection	9.1	9.9	12.6

Table 4-5: Percentages of Simultaneities that are Novelties

Category	DKS	FSFO	SBFS
Class Complex Subsets	7	37	35
% Change	114.3	62.2	20
Class Collection Subsets	15	60	42
Class Complex Supersets	13	80	102
% Change	130.8	71.3	363.7
Class Collection Supersets	30	137	473

Table 4-6: Subsets & Supersets

Sequences

In considering the sequences of simultaneities, it is similarly helpful to assess the overall trends between pieces. Table 4-7 provides the total counts for different simultaneities in each piece. Not only are the pieces themselves clearly highly variegated, but so are the disparities between the profiles of the different pieces. Particularly notable is the difference between FSFO & SBFS, especially in the context of Table 4-2: whilst they have approximately the same number of different pitch complexes, SBFS has almost twice as many different types of sequence constructed from these, despite the two works having roughly the same duration. Nonetheless, they have much more similar numbers of class complexes and class collections. The similarity in class complexes combined with the disparity in pitch complexes indicates much greater variety of transposition in SBFS: whilst the two have roughly as many sequences of 'types' of simultaneity, these are employed to make many more transposed sequences in SBFS. As for DKS, whilst some class complex sequences are clearly reused at different transposition levels, there is hardly any reuse of the same class collection sequences with different voicings.

Piece	DKS	FSFO	SBFS	DKS	FSFO	SBFS
Sequence Length	2			3		
Pitch Complex	102	375	714	108	425	746
% Change	-25.5	-20.3	-61.6	-15.7	-16.5	-55.6
Class Complex	76	299	274	91	355	331
% Change	-5.3	-10.4	-7.3	-2.2	-3.9	-4.8
Class Collection	72	268	254	89	341	315

Table 4-7: Total Sequences

The overall trends for mutual sequences (Table 4-8) are largely as expected: as the level of reduction increases, there are greater numbers of mutual sequences. One feature that requires explication is the decline in numbers between the class complex and class collection count for DKS & FSFO: this is because some of these sequences start occurring in all three works at the class collection level. That the figures for FSFO & SBFS are always the highest is hardly surprising, as they have the greatest number of sequences; particularly notable is the number of 2-simultaneity class collection sequences, which is approximately 10% of the total number of sequences in each piece. Nonetheless, that this is by far the highest figure demonstrates fundamentally how rare mutual sequences are.

Indeed, through examining these sequences, it becomes clear that, as before, those simultaneities that are shared between pieces are rudimentary. Monads are very prevalent throughout, whilst larger simultaneities tend to be a feature only at lower levels of reduction. In fact, even at the greatest level of harmonic reduction (i.e. class collections), those sequences that recur in all three works are very basic, and all employ (0) as at least one of the entities. Likewise, any sequence that recurs in any combination of pieces with a tetradic simultaneity is alongside (0) or (0) and (0), depending on the length of sequence. Thus, the 'largest' sequences that occur are triads with dyads, which only occur in FSFO & SBFS: (0 5) (0 1 6), (0 3 11) (0 11), (0 4) (0 3 4), and (0) (0 5) (0 1 6). Even these are hardly hugely recognisable: all are related either by the addition or subtraction of a pitch class, or through semitonal voice leading, or both. Mutual sequence novelties display the same trends: this is a very rare phenomenon, and when it does occur it is with small entities.

Piece	DKS & FSFO	DKS & SBFS	FSFO & SBFS	All Three	DKS & FSFO	DKS & SBFS	FSFO & SBFS	All Three
Sequence Length	2				3			
Pitch Complex	0	0	1	0	0	0	0	0
% Change	N/A	N/A	600.0	N/A	N/A	N/A	N/A	N/A
Class Complex	5	6	7	4	1	4	5	3
% Change	-40.0	33.3	228.6	100.0	0.0	50.0	100.0	33.3
Class Collection	3	8	23	8	1	6	10	4

Table 4-8: Mutual Sequences

As for novelties themselves, these are more varied (Table 4-9). FSFO, with much higher percentages, is somewhat anomalous here, indicating repetition of sequences as much more common in this work. SBFS and DKS have greater similarities in this regard, although there is a higher change for SBFS.

Piece	DKS	FSFO	SBFS	DKS	FSFO	SBFS
Sequence Length	2			3		
Pitch Complex	1.0	12.8	1.1	0.0	10.8	0.4
% Change	436.8	17.6	844.6	N/A	1.5	1402.5
Class Complex	5.3	15.1	10.6	3.3	11.0	6.1
% Change	111.1	33.9	56.2	70.4	9.4	31.3
Class Collection	11.1	20.1	16.5	5.6	12.0	7.9

Table 4-9: Percentages of Sequences that are Novelties

Table 4-10 shows two features of those sequences that are novelties: whether their constituent simultaneities display maximum connectivity or no or one note difference between them. The profiles for the three pieces are variable, although there are some common trends. The consistently high figures for no or one note change indicate that those sequences which are reused (i.e. are novelties) consist of simultaneities which are very similar, with very few notes changing; nonetheless, contrasting these figures to those of maximum connectivity indicates that direct repetition is comparatively rare, and indeed the note that does change tends to do so by a relatively large amount.

Percentage of Novelties with	Maximum Connectivity			No or One Note Difference		
Piece	DKS	FSFO	SBFS	DKS	FSFO	SBFS
Pitch Complex	0	15	21	N/A	79	100
% Change	50	9	8	N/A	3	-9
Class Complex	50	24	29	100	82	91
% Change	-6	-14	-6	-6	0	-29
Class Collection	44	10	23	94	82	62

Table 4-10: Features of Novelty Sequences

Case Study: *Sechs Bagatellen Für Streichquartett*, Op. 9

This chapter concludes with a case study on SBFS. In part due to economy of space, and in part given the intention of this thesis, this should not be regarded as an exhaustive or comprehensive analysis of this work; rather, it uses the data gathered above and the resulting phenomena as a way of looking at this particular work in some more detail, and in comparison to some of the other analytical work that has been carried out on this piece. Indeed, the decision to use this work for the case study was in part due to the wealth of analytical scholarship that relates to it. Again, this case study must inevitably forgo detailed examination of all of the existing scholarship, but instead it will identify some of the major trends in analysis of this work and consider how this sort of data-driven approach can critique such thought.

Most of the formalist analytical work that has been carried out on this piece has been – to a greater or lesser extent – neo-Fortean. There is a body of work exploring several features which have been considered as somehow crucial to the pitch-structuring of the piece. Perhaps the most frequently discussed of these refers to the idea of the 'run', a concept that Webern introduced in his own discussion of this work in 1932 (Webern, 1963, p. 51). As Robert Harry Hallis Jr. discusses, this is essentially the unfolding of the total chromatic, although runs can be shorter than all 12 pitch classes and tend to occur within a section of music marked out as structurally significant through other parameters. Crucially, pitch repetition does occur, and so although it can be seen as a precursor to serialism, it is by no means synonymous with it (Hallis Jr., 2004, pp. 3–4). Nonetheless, Webern's proclamation of the relevance of the 'run' to SBFS has led several authors to consider the relevance of chromatic completion as a structural device in this work (Chrisman, 1979; Hallis Jr., 2004, pp. 346–358; Paccione, 1988). An alternative approach, though similarly linked to ideas of the total chromatic, is that of an expanding chromatic wedge creating a series of dyads which form the basic generative idea of the work. For Davies (2007) this has no motivic expression, but is rather a background structural force; similarly, Pearsall (1991) considers the relevance of structural dyads and their filling-in as an analogy to background harmonic prolongation. More broadly, Chrisman (1979) draws attention to the extensive use of semitonal relationships throughout the work, often on the surface level, as well as symmetrical features in the harmony. Finally, in his own analysis of the work Forte (1998, pp. 169–203) locates a pervasive octatonic presence in the pitch class set content.

Some of these features lend themselves to interrogation in the light of this thesis more readily than others: as the present study engages deliberately with the surface of the music, consideration of background structures is feasible only in the light of their surface presentation; likewise, although a data-driven approach could be developed to consider the precise ordering of pitch elements across the exposition of the total chromatic, it would be different from the one employed here.

Instead, those ideas which do lend themselves to interrogation through the lens of this thesis are those with a concern for the localised pitch collections on the surface of the music. The broadest of these is the presence of semitonal relations as a characteristic feature of this work. The most obvious way of considering this is to ask how many simultaneities include at least one semitone: out of 166 different class complexes, only 24 (14.5%) include a semitone; however, out of 88 different class collections – closer to the pitch class sets that many of the above authors are considering – 55 include a semitone (62.5%).[1] Similarly, four of the 13 class complex novelties and seven of the 11 class collection novelties include a semitone. Considering the other group of simultaneities with particular importance, subsets that recur as parts of supersets, three out of 35 class complex subsets include a semitone, but 31 out of 42 class collection subsets do. The disparities here between class collections and class complexes are highly revealing. Whilst semitones appear frequently in class collections, affirming the basic argument of previous authors who have largely considered the pitch class set as the harmonic unit – an even greater reduction than the class collection – they are much less common in class complexes. This implies that whilst they are a characteristic feature of Webern's pitch collections, he rarely literally exposes them in the voicing of the pitches in the work. This study does not consider horizontal, contrapuntal pitch content, but the observation is nonetheless important and clear.

[1] In this discussion compound intervals are assumed when discussing class collections, but not when discussing class complexes. Thus, the pitch complex (D4 A5 A#6) does not include a semitone when viewed as class complex (C4 G5 G#6) but does when viewed as class collection (0 7 8).

Extending this idea further, Chrisman draws particular attention to three-note units of a semitone with a further interval, and four-note symmetrical units of the form semitone-interval-semitone, and goes on to argue that many larger sets are associated with these smaller ones (i.e. are supersets of them) (Chrisman, 1979, pp. 84–85). Considering these in the context of class collection novelties in this work, two out of three triads fit this trend, as do both tetrads; likewise 20 out of 31 triad subsets, but only three out of 11 tetrad subsets.

In a similar manner, in his discussion of the second movement of SBFS, Pearsall argues that semitones have a function something analogous to a consonance, and thus that the (0 2) dyad (again, he is referring to pitch class sets) requires 'filling in' to be (0 1 2), with the exception of its appearance as (0 1 3 4) (Pearsall, 1991, pp. 353–354). It is notable, therefore, that the class collection (0 1 3 4) never occurs in *any* movement in this piece, nor do its possible rearrangements ((0 2 3 11), (0 1 9 10), or (0 8 9 11)). Similarly, (0 2) is never followed by (0 1 2) (it is once followed by (0 1), and once by (0 2 3)). That said, Pearsall's point is more structural and less concerned with the surface pitch-material, and so perhaps more significant is the occurrence of both the dyad (0 2) and the triads (0 1 2) as novelties.

The last proposition under consideration here is Forte's identification of an octatonic character in the harmony, it is again possible to consider how many simultaneities of varying significance fit into an octatonic collection. Given Forte's adherence to pitch class sets, the following discussion will consider only class collections (not quite the same, but very close). Here, 38 out of 88 total class collections fit into an octatonic collection, as do eight out of 11 novelties, although only two out of five larger novelties (greater than a dyad) do. Finally, 21 out of 42 of the subsets that form a superset fit into the octatonic collection. These numbers are not insignificant, although they do not form a majority, and so although Forte's proposition cannot be dismissed, it appears to be limited in relevance. Indeed, it is worth noting that many of these octatonic units are small: the monad collection and all dyads – 12 of the 38 – fit into an octatonic collection. These are hardly distinctive entities, and so leave 26 out of the 76 larger entities as octatonic – still a meaningful proportion, but not as high, and nowhere near a majority.

Conclusion

To conclude the thesis, this chapter returns to the opening research questions. By marshalling the trends across the pieces, as well as the more idiosyncratic features identified above, the answers to these questions can concern not only broad patterns in Webern's syntactical praxis, but also how these pieces differ. Before answering the questions posed above, however, this chapter will proffer some potential areas of future study, building on this method of data-based analysis.

Further Study

One way to build on this approach would be to consider a wider body of work. By contemplating more pieces of Webern from a similar period, it is possible to have a much wider sense of the degree to which his music was coherent and unified in its pitch content: which features are consistent across different works, and which are more variable. More broadly, this sort of technique could also be used to compare the profiles of works from a particular stylistic group to others: Webern's 'freely atonal' works could be compared to his serial ones, or to Schoenberg's 'freely atonal' works. In a more detailed manner, it could be highly rewarding, for example, to chart the change in Webern's orchestral writing from his *Sechs Stücke für grosses Orchester,* Op. 6 to his *Variationen für Orchester,* Op. 30, via the *Fünf Stücke für Orchester,* Op. 10, discussed here, and the *Symphonie,* Op. 21. There is no reason that this need confine itself to pitch content, either. Analysis of more advanced textural matters could certainly be considered, alongside rhythmic or dynamic features.

From the other side of the enquiry, it would be possible to use just the data in this thesis in a more specifically targeted manner to consider particular aspects of his music. The case study above gives an initial indication of how this might be achieved, but by using more sophisticated statistical models, Markov chains are a common strategy, for example (e.g. Jacoby et al., 2015; Raphael & Stoddard, 2004), it would be possible not only to find more subtle patterns, but also to answer different types of question, and to interrogate common assumptions in analytical scholarship that imply an empirical basis, but have not been subjected to rigorous enquiry.

Finally, these tools are potentially applicable to totally different styles of music, with some adaptation. Corpus study projects are still in their comparative infancy in musicology, particularly in relation to twentieth-century repertoire, where the extreme heterogeneity of the repertoire can imply that finding legitimate corpora is a difficult, if not impossible task. Nonetheless, as this thesis has demonstrated, a corpus study approach not only is possible, but can be extremely enlightening, even if considering a comparatively small corpus.

Research Questions Revisited

The first question concerned whether particular types of simultaneity are used more frequently than others in these works. The answer to this is somewhat multifaceted: whilst certain simultaneities occur more frequently within each work, and indeed some occur moderately frequently in all three works, these tend to be small, fairly simple, foundational units, typically monadic or dyadic. Thus, although Webern clearly has a well from which he can draw basic 'types' of simultaneity, there is no significant reuse of complex pitch collections between pieces, and although some occur in various places in a piece (particularly in SBFS), this is also rare, and typically attention is not drawn to it in context.

The second question asked the same of sequences of simultaneities, and the situation here is even more diverse. Any repetition of a sequence of simultaneities is notable in itself, as Webern strives to avoid this. As with simultaneities, the only sequences that recur across pieces tend to be very reduced, with little in terms of noticeable replicated voice-leading patterns. Likewise, persistent repetition within pieces is also rare: typically, it occurs as the result of a repetitive texture, rather than through similar patterns recurring in different locales of a piece. Indeed, these repetitive textures are often essentially harmonically static, and so any sense of motion provided by the texture is counteracted by the pitch content.

From these conclusions, it is possible to infer, to a degree, Webern's aims with his newly-deployed pitch material. Clearly, repetition was to be avoided as a general rule. In 1932, describing the process by which he and his compatriots stumbled towards serialism, he wrote that 'an idea occurred to us: "We don't want to repeat, there must constantly be something new!"' (Webern, 1963, p. 55). Clearly this had been an aim, whether conscious or not, as early as 1911.

Indeed, the patterns of pitch classes above demonstrate how successfully he avoided prioritising any pitch class in any of these pieces, thereby avoiding any possible tonicisation (at least through repetition). This avoidance of repetition applies to simultaneities too, even if there are evidently differences in his approach in these three pieces: transposition of the same collections is much more common in SBFS than the other two works; likewise, the piece is less texturally sparse than the other two works. Although they all have their idiosyncrasies, SBFS can be considered to be something of an outlier. It therefore does not seem that Webern was attempting to establish a new universal 'language' in an analogous way to that provided by tonality – i.e. a 'vocabulary' of types of simultaneity, which are deployed in syntactical sequences – but rather that he sought to distinguish each piece by its own pitch content. Pearsall wrote that 'It is possible to imagine a universe of post-tonal compositions where each composition defines its own harmonic structures and pitch hierarchy' (Pearsall, 1991, p. 347). No longer must we imagine.

Bibliography

Aarden, B. J. (2003). *Dynamic Melodic Expectancy*. The Ohio State University.

Biber, D., Reppen, R., & Friginal, E. (2012). Research in Corpus Linguistics. In R. B. Kaplan (Ed.), *The Oxford Handbook of Applied Linguistics, (2 Ed.)* (2nd ed.). https://doi.org/10.1093/oxfordhb/9780195384253.013.0038

Bonte, T., Froment, N., & Schweer, W. (n.d.). MuseScore. Retrieved May 18, 2019, from https://musescore.com

Boulez, P. (1968). *Notes of an Apprenticeship* (P. Thévenin, Ed.; H. Weinstock, Trans.). New York: Alfred A. Knopf.

Butterfield, A., & Ekembe Ngondi, G. (Eds.). (2016). Open-Source. In *Oxford Dictionary of Computer Science* (7th ed.). https://doi.org/10.1093/acref/9780199688975.001.0001

Chrisman, R. (1979). Anton Webern's "Six Bagatelles for String Quartet," Op. 9: The Unfolding of Intervallic Successions. *Journal of Music Theory*, *23*(1), 81–122. https://doi.org/10.2307/843695

Cook, N. (2004). Computational and Comparative Musicology. In E. Clarke & N. Cook (Eds.), *Empirical Musicology: Aims, Methods, Prospects* (pp. 103–126). Oxford: Oxford University Press.

Cuthbert, M. S., & Ariza, C. (2010). Music21 A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In J. S. Downie & R. C. Veltkamp (Eds.), *Proceedings of the 11th International Society for Music Information Retrieval Conference*. Utrecht: ISMIR.

Cuthbert, M. S., Hadley, B., Johnson, L., & Reyes, C. (2012). Interoperable Digital Musicology Research via music21 Web Application. *Joint CLARIN-D/DARIAH Workshop at Digital Humanities Conference Hamburg*.

Davies, B. K. (2007). The structuring of tonal space in Webern's six bagatelles for string quartet, Op. 9. *Music Analysis*, *26*(1/2), 25–58. https://doi.org/10.1111/j.1468-2249.2008.00273.x

Debiasi, G. B., & de Poli, G. G. (1982). Musica (musicae usitata scriptura idonee calculatoribus aptata): A language for the transcription of musical texts for computers. *Journal of New Music Research*, *11*(1), 1–27. https://doi.org/10.1080/09298218208570342

Dipert, R. R. (1977). Allen Forte – "The Structure of Atonal Music" (Review). *Indiana Theory Review*, *1*(1), 5–11.

Downie, J. S. (2004). The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future. *Computer Music Journal*, *28*(2), 12–23. https://doi.org/10.1162/014892604323112211

ELVIS Project. (n.d.). Retrieved May 18, 2019, from https://elvisproject.ca

Forte, A. (1973). *The Structure of Atonal Music*. New Haven & London: Yale University Press.

Forte, A. (1986). Letter to the Editor in Reply to Richard Taruskin from Allen Forte. *Music Analysis*, *5*(2/3), 321–337. https://doi.org/10.2307/854194

Forte, A. (1998). *The Atonal Music of Anton Webern*. New Haven & London: Yale University Press.

Forte, A. (2006). Sets and Nonsets in Schoenberg's Atonal Music. *Perspectives of New Music*, *11*(1), 43–64. https://doi.org/10.2307/832462

Fuller, R. (1970). Toward a Theory of Webernian Harmony, via Analysis with a Digital Computer. In H. B. Lincoln (Ed.), *The Computer and Music* (pp. 123–131). Ithaca, NY: Cornell University Press.

Guo, E. W. (n.d.). IMSLP: Petrucci Music Library. Retrieved May 18, 2019, from https://imslp.org

Hallis Jr., R. H. (2004). *Reevaluating the Compositional Process of Anton Webern: 1910-1925*. The University of Texas at Austin.

Hasty, C. (1981). Segmentation and Process in Post-Tonal Music. *Music Theory Spectrum*, *3*, 54–73. https://doi.org/10.2307/746134

Jackson, R. (1970). Harmony before and after 1910: A Computer Comparison. In H. B. Lincoln (Ed.), *The Computer and Music* (pp. 132–146). Ithaca, NY: Cornell University Press.

Jacoby, N., Tishby, N., & Tymoczko, D. (2015). An Information Theoretic Approach to Chord Categorization and Functional Harmony. *Journal of New Music Research*, *44*(3), 219–244. https://doi.org/10.1080/09298215.2015.1036888

Janssens, H., & Landrieu, W. (1976). Melowriter, a digital music coding machine. *Journal of New Music Research*, *5*(4), 225–247. https://doi.org/10.1080/09298217608570226

Klein, E., & Silver, N. (2018, October 23). The Ezra Klein Show: What Nate Silver's learned about forecasting elections. Retrieved June 14, 2019, from Vox.com website: https://www.vox.com/ezra-klein-show-podcast/2018/10/23/18014156/nate-silver-538-forecasting-2018-2020-ezra-klein-podcast

Lewin, D. (1983). Transformational Techniques in Atonal and Other Music Theories. *Perspectives of New Music*, *21*(1/2), 312–371. https://doi.org/10.2307/832879

Lewin, D. (2010). Set Theory, Derivation, and Transformational Structures in Analyzing Webern's Opus 10, Number 4. In *Musical Form and Transformation: Four Analytic Essays*. https://doi.org/10.1093/acprof:oso/9780195317121.003.0003

Mesnage, M. (1993). Morphoscope, a computer system for music analysis. *Journal of New Music Research*, *22*(2), 119–131. https://doi.org/10.1080/09298219308570624

Moreno, R. R. (2017). Harmonic Syntax and Vocabulary in Tonal Music. *9th European Music Analysis Conference*.

Paccione, P. (1988). Chromatic Completion: Its Significance in Tonal and Atonal Contexts. *College Music Symposium*, *28*, 85–93.

Pearsall, E. R. (1991). Harmonic Progressions and Prolongation in Post-Tonal Music. *Music Analysis*, *10*(3), 345–355. https://doi.org/10.2307/853972

Pugin, L. (2015). The Challenge of Data in Digital Musicology. *Frontiers in Digital Humanities*, *2*(4). https://doi.org/10.3389/fdigh.2015.00004

Raphael, C., & Stoddard, J. (2004). Functional Harmonic Analysis Using Probabilistic Models. *Computer Music Journal*, *28*(3), 45–52. https://doi.org/10.1162/0148926041790676

Rodin, J., Sapp, C. S., & Bokulich, C. (n.d.). The Josquin Research Project. Retrieved May 18, 2019, from http://josquin.stanford.edu

Rohrmeier, M., & Cross, I. (2008). Statistical Properties of Tonal Harmony in Bach's Chorales. In K. Miyazaki (Ed.), *Proceedings of the 10th International Conference on Music Perception and Cognition*. Sapporo: ICMPC.

Roig-Francoli, M. A. (2001). A Theory of Pitch-Class-Set Extension in Atonal Music. *College Music Symposium*, *41*, 57–90.

Rootham, D., Jonas, P., & Gotham, M. (n.d.). Lieder Corpus Project. Retrieved May 18, 2019, from https://fourscoreandmore.org/scores-of-scores/lieder-corpus-project/

Solomon, L. (1982). The list of chords, their properties and use in analysis. *Journal of New Music Research*, *11*(2), 61–107. https://doi.org/10.1080/09298218208570346

Straus, J. N. (2006). The Problem of Prolongation in Post-Tonal Music. *Journal of Music Theory*, *31*(1), 1–21. https://doi.org/10.2307/843544

Taruskin, R. (1979). Allen Forte – The Harmonic Organization of The Rite of Spring (Review). *Current Musicology*, (28), 114–129.

Taruskin, R. (1986). Letter to the Editor from Richard Taruskin. *Music Analysis*, *5*(2/3), 313–320. https://doi.org/10.2307/854193

VIS Framework for Music Analysis. (n.d.). Retrieved May 12, 2019, from https://vis-framework.readthedocs.io/en/v3.0.5/#

Webern, A. (1923). *Fünf Stücke für Orchester*. Vienna: Universal Edition.

Webern, A. (1924a). *Drei Kleine Stücke, Op. 11*. Vienna: Universal Edition.

Webern, A. (1924b). *Sechs Bagatellen für Streichquartett, Op. 9*. Vienna: Universal Edition.

Webern, A. (1963). *The Path to the New Music* (W. Reich, Ed.; L. Black, Trans.). Bryn Mawr, Pnnsylvania: Theodore Presser Company.

Weiß, C., Balke, S., Abeßer, J., & Müller, M. (2018). Computational Corpus Analysis: a Case Study on Jazz Solos. In E. Gómez, X. Hu, E. Humphrey, & E. Bennetos (Eds.), *Proceedings of the 19th International Society for Music Information Retrieval* (pp. 416–423). Paris, France: ISMIR.

White, C. W. (2013). *Some Statistical Properties of Tonality, 1650–1900*. Yale University.

Yust, J. (2019). Stylistic information in pitch-class distributions. *Journal of New Music Research*. https://doi.org/10.1080/09298215.2019.1606833

Appendix I

Notes

Below is the code used for data-gathering, written in Python, employing the music21 toolkit. Basic knowledge of Python is required for reading the details of the code, but it is hoped that the annotations will allow any reader to have a sense of the manner in which the code functions. Some basic forms which recur frequently are given below to aid the reader.

The code is split into 9 sections: lines 1-18 define some preliminary functions; lines 20-71 gather data about pitch complexes; 73-128 about class complexes; 130-217 about class collections; 219-276 about sequences of pitch complexes; 278-343 about sequences of class complexes; 345-442 about sequences of class collections; 444-541 about pitches; and 543-633 about pitch classes. The code below only covers 2-simultaneity sequences as the code for 3-simultaneity sequences is functionally identical, and so does not require exposition here. As there is lots of overlap between different code sections that find different levels of harmonic reduction, only the differing lines of code have been explained.

Throughout the coding, 'chord' has been used as an equivalent for 'simultaneity', largely due to the comparative length of the two words, but also because once the piece is chordified, music21 understands each simultaneity as a chord.

Glossary

def *abc*(*d*) : define a function *abc* with variable *d*

efg = [] : define an empty list named *efg*

for *h* in *I* : for the element *h* in the list of elements *i*

<table>
<tr><th>Line</th><th>Code</th><th>Comments</th></tr>
<tr><td>1</td><td>def attempt4(string):</td><td></td></tr>
<tr><td>2</td><td>return string.replace("music21.pitch.Pitch ", "").replace(">", "").replace("<", "").replace(",", "").replace("((", "(").replace("))", ")")</td><td>For a given string, replaces "x" with "y"</td></tr>
<tr><td>3</td><td></td><td></td></tr>
<tr><td>4</td><td>csvOut=open("/Users/joshua ballance/Desktop/newtest.csv", "w")</td><td rowspan="4">Creates a new CSV file with these entries in the first row</td></tr>
<tr><td>5</td><td>titleWriter = csv.writer(csvOut, delimiter=',', quotechar='|')</td></tr>
<tr><td>6</td><td>titleWriter.writerow(['Filename, Category, Chord, Chord Length, Total Duration, Number of Times Struck'])</td></tr>
<tr><td>7</td><td>csvOut.close()</td></tr>
<tr><td>8</td><td></td><td></td></tr>
<tr><td>9</td><td>def csvWriter():</td><td rowspan="5">Opens the CSV file and writes subsequent rows with these variables</td></tr>
<tr><td>10</td><td>csvOut=open("/Users/joshua ballance/Desktop/newtest.csv", "a")</td></tr>
<tr><td>11</td><td>barlineWriter = csv.writer(csvOut, delimiter=',', quotechar='|')</td></tr>
<tr><td>12</td><td>barlineWriter.writerow([fileTitle, dataCategory, chordName, chordLength, totalDuration, numberOfTimesStruck])</td></tr>
<tr><td>13</td><td>csvOut.close()</td></tr>
<tr><td>14</td><td></td><td></td></tr>
<tr><td>15</td><td>pieceName = []</td><td rowspan="3">Creates a list of filepaths for each piece</td></tr>
<tr><td>16</td><td>pieceName.append('~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/Drei Kleine Stücke SOUNDING.mxl')</td></tr>
<tr><td>17</td><td>pieceName.append('~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/Fünf Stücke für Orchester SOUNDING DRAFT 1 FULL.mxl')</td></tr>
</table>

18	pieceName.append('~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/Sounding/Sechs Bagatellen für Streichquartett SOUNDING FULL.mxl')	
19		
20	for el in pieceName:	For each piece
21	thePiece = converter.parse(el)	Parses the file so it is readable by music21
22	fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")	Defines the fileTitle variable with the title of the piece
23	chordifiedPiece = thePiece.chordify()	Chordifies the piece
24	listOfChords = []	
25	chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')	Gets each 'chord element' (i.e. every simultaneity)
26	for el in chords:	Creates a list of chords
27	listOfChords.append(el)	
28	newListOfChords = []	
29	for el in listOfChords:	For the list of chords, removes any that are at the end of a tie or the middle of a tie, if the pitches are the same as the previous chord
30	positionInList = listOfChords.index(el)	
31	if 'stop' in str(el.tie):	
32	if el.pitches == listOfChords[positionInList-1].pitches:	

pass	
else:	
newListOfChords.append(el.pitches)	
else:	
if 'continue' in str(el.tie):	
if el.pitches == listOfChords[positionInList-1].pitches:	
pass	
else:	
newListOfChords.append(el.pitches)	
else:	
newListOfChords.append(el.pitches)	
listOfTransposedChords = []	
for el in chords:	
listOfTransposedChords.append(el)	
listOfTransposedChordsPitches = []	
for el in listOfTransposedChords:	
listOfTransposedChordsPitches.append(el.pitches)	
orderedListOfIndividualTranposedChordsPitches = sorted(list(set(listOfTransposedChordsPitches)), key=len)	Creates a list of each different chord (by pitch content) in the piece
listOfDurations = []	
for el in orderedListOfIndividualTranposedChordsPitches:	For each chord in the list of different chords in the piece,
listOfInitialDurations = []	

54	for i in listOfTransposedChords:	creates a sublist with the chord and its total duration in seconds
55	if i.pitches == el:	
56	listOfInitialDurations.append(i.seconds)	
57	listOfSummedDurations = []	
58	listOfSummedDurations.append(el)	
59	listOfSummedDurations.append(sum(listOfInitialDurations))	
60	listOfDurations.append(listOfSummedDurations)	
61	orderedList = sorted(listOfDurations, key=itemgetter(1))	
62	tidyListOfDurations = []	
63	for el in orderedList:	
64	tidyListOfDurations.append(el)	
65	for el in tidyListOfDurations:	Appends the relevant data to the relevant variable, and writes it to the CSV file
66	dataCategory = 'Chord (Pitches)'	
67	chordName = attempt4(str(el[0]))	
68	chordLength = len(el[0])	
69	totalDuration = str(el[1])	
70	numberOfTimesStruck = 'N/A'	
71	csvWriter()	
72		
73	for el in pieceName:	
74	thePiece = converter.parse(el)	
75	fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil	

	Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")	
76	chordifiedPiece = thePiece.chordify()	
77	listOfChordsWithDynamics = []	
78	chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')	
79	listOfTransposedChords = []	
80	for el in chords:	Creates a list with each chord transposed to have C4 as its root note
81	rootOfChord = el.bass()	
82	middleC = note.Note('c4')	
83	rootNote = note.Note(rootOfChord)	
84	transpositionInterval = interval.notesToChromatic(rootNote, middleC)	
85	transpositionIntervalSemitones = transpositionInterval.semitones	
86	transposedChord = el.transpose(transpositionIntervalSemitones)	
87	listOfTransposedChords.append(transposedChord)	
88	listOfTransposedChordsPitches = []	
89	for el in listOfTransposedChords:	
90	listOfTransposedChordsPitches.append(el.pitches)	
91	newListOfChords = []	
92	for el in listOfTransposedChords:	
93	positionInList = listOfTransposedChords.index(el)	
94	if 'stop' in str(el.tie):	

```
            if el.pitches == listOfTransposedChords[positionInList-1].pitches:
                pass
            else:
                newListOfChords.append(el.pitches)
        else:
            if 'continue' in str(el.tie):
                if el.pitches == listOfTransposedChords[positionInList-1].pitches:
                    pass
                else:
                    newListOfChords.append(el.pitches)
            else:
                newListOfChords.append(el.pitches)
    orderedListOfIndividualTranposedChordsPitches = sorted(list(set(listOfTransposedChordsPitches)),
key=len)
    listOfDurations = []
    for el in orderedListOfIndividualTranposedChordsPitches:
        listOfInitialDurations = []
        for i in listOfTransposedChords:
            if i.pitches == el:
                listOfInitialDurations.append(i.seconds)
        listOfSummedDurations = []
        listOfSummedDurations.append(el)
        listOfSummedDurations.append(sum(listOfInitialDurations))
```

```
        listOfDurations.append(listOfSummedDurations)
    orderedList = sorted(listOfDurations, key=itemgetter(1))
    tidyListOfDurations = []
    for el in orderedList:
        tidyListOfDurations.append(el)
    for el in tidyListOfDurations:
        dataCategory = 'Chord Type (Pitches)'
        chordName = attempt4(str(el[0]))
        chordLength = len(el[0])
        totalDuration = str(el[1])
        numberOfTimesStruck = 'N/A'
        csvWriter()

for el in pieceName:
    thePiece = converter.parse(el)
    fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")
    chordifiedPiece = thePiece.stripTies().chordify()
    listOfChordsWithDynamics = []
```

chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')	
listOfTransposedChords = []	
for el in chords:	
rootOfChord = el.bass()	
middleC = note.Note('c4')	
rootNote = note.Note(rootOfChord)	
transpositionInterval = interval.notesToChromatic(rootNote, middleC)	
transpositionIntervalSemitones = transpositionInterval.semitones	
transposedChord = el.transpose(transpositionIntervalSemitones)	
listOfTransposedChords.append(transposedChord)	
listOfTransposedChordsPitches = []	
for el in listOfTransposedChords:	Creates a new list of each chord represented by its pitch class content
listOfTransposedChordsPitches.append(el.orderedPitchClassesString)	
newListOfChords = []	
for el in listOfTransposedChords:	
positionInList = listOfTransposedChords.index(el)	
if 'stop' in str(el.tie):	
if el.pitches == listOfTransposedChords[positionInList-1].pitches:	
pass	
else:	
newListOfChords.append(el)	
else:	

157	if 'continue' in str(el.tie):	
158	if el.pitches == listOfTransposedChords[positionInList-].pitches:	
159	pass	
160	else:	
161	newListOfChords.append(el)	
162	else:	
163	newListOfChords.append(el)	
164	secondListOfChords = []	
165	for el in newListOfChords:	
166	secondListOfChords.append(el.orderedPitchClassesString)	
167	orderedListOfIndividualTranposedChordsPitches = sorted(list(set(listOfTransposedChordsPitches)), key=len)	
168	listOfDurations = []	
169	for el in orderedListOfIndividualTranposedChordsPitches:	
170	listOfInitialDurations = []	
171	for i in listOfTransposedChords:	
172	if i.orderedPitchClassesString == el:	
173	listOfInitialDurations.append(i.seconds)	
174	listOfSummedDurations = []	
175	listOfSummedDurations.append(el)	
176	listOfSummedDurations.append(sum(listOfInitialDurations))	
177	listOfDurations.append(listOfSummedDurations)	
178	orderedList = sorted(listOfDurations, key=itemgetter(2))	

```
tidyListOfDurations = []
for el in orderedList:
    tidyListOfDurations.append(el)
temporaryListOfData = []
for el in tidyListOfDurations:
    newChordName = []
    for i in el[0]:
        if i == '<':
            pass
        else:
            if i == '>':
                pass
            else:
                if i == 'A':
                    newChordName.append('10')
                else:
                    if i == 'B':
                        newChordName.append('11')
                    else:
                        newChordName.append(i)
    newChordData = []
    newChordData.append(newChordName)
    newChordData.append(el[1])
```

Tidies up the form of the pitch class string, making it more easily readable

```
        temporaryListOfData.append(newChordData)
    finalListOfData = []
    for el in temporaryListOfData:
        temporaryFinalData = []
        i = str(el[0]).replace("'", "").replace("[", "").replace("]", "").replace(",", "")
        temporaryFinalData.append(i)
        temporaryFinalData.append(len(el[0]))
        temporaryFinalData.append(el[1])
        finalListOfData.append(temporaryFinalData)
    for el in finalListOfData:
        dataCategory = 'Chord Type (Pitch Classes)'
        chordName = el[0]
        chordLength = el[1]
        totalDuration = str(el[2])
        numberOfTimesStruck = 'N/A'
        csvWriter()

for el in pieceName:
    thePiece = converter.parse(el)
    fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine
Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil
Corpus/Fünf Stücke für Orchester/Sounding/",
"").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für
```

```
Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL",
"").replace(".mxl", "")
    chordifiedPiece = thePiece.chordify()
    listOfChordsWithDynamics = []
    chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')
    listOfTransposedChords = []
    for el in chords:
        listOfTransposedChords.append(el)
    firstListOfPositions = list(range(0, len(listOfTransposedChords)))
    newListOfChords = []
    for el in firstListOfPositions:
        if 'stop' in str(listOfTransposedChords[el].tie):
            if listOfTransposedChords[el].pitches == listOfTransposedChords[el-1].pitches:
                pass
            else:
                newListOfChords.append(listOfTransposedChords[el])
        else:
            if 'continue' in str(listOfTransposedChords[el].tie):
                if listOfTransposedChords[el].pitches == listOfTransposedChords[el-1].pitches:
                    pass
                else:
                    newListOfChords.append(listOfTransposedChords[el])
            else:
```

243	newListOfChords.append(listOfTransposedChords[el])	
244	listOfTransposedChordSequences = []	
245	listOfPositions = list(range(0, len(newListOfChords)))	
246	for el in listOfPositions:	Creates a list of all two-chord sequences
247	if el < len(newListOfChords)-1:	
248	newChordSequence = []	
249	newChordSequence.append(newListOfChords[el])	
250	newChordSequence.append(newListOfChords[el+1])	
251	listOfTransposedChordSequences.append(newChordSequence)	
252	listOfChordSequencePitches = []	
253	for el in listOfTransposedChordSequences:	Creates a list of these sequences just including pitch content
254	i = el[0].pitches, el[1].pitches	
255	listOfChordSequencePitches.append(i)	
256	listOfDurations = []	
257	results3 = Counter()	Counts each statement of this sequence in the list of sequences
258	for el in listOfChordSequencePitches:	
259	results3[el] = listOfChordSequencePitches.count(el)	
260	listOfSummedDurations = []	
261	listOfSummedDurations.append(el)	
262	listOfSummedDurations.append(results3[el])	
263	listOfDurations.append(listOfSummedDurations)	
264	tidyListOfDurations = []	
265	for el in listOfDurations:	

```
        if el in tidyListOfDurations:
            pass
        else:
            tidyListOfDurations.append(el)
    for el in tidyListOfDurations:
        dataCategory = 'Chord Sequence (2 Chords - Chords - Pitches)'
        chordName = attempt4(str(el[0]))
        chordLength = 'N/A'
        totalDuration = 'N/A'
        numberOfTimesStruck = str(el[1])
        csvWriter()

for el in pieceName:
    thePiece = converter.parse(el)
    fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")
    chordifiedPiece = thePiece.stripTies().chordify()
    chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')
    firstListOfChords = []
```

```
for el in chords:
    firstListOfChords.append(el)
firstListOfPositions = list(range(0, len(firstListOfChords)))
newListOfChords = []
for el in firstListOfPositions:
    if 'stop' in str(firstListOfChords[el].tie):
        if firstListOfChords[el].pitches == firstListOfChords[el-1].pitches:
            pass
        else:
            newListOfChords.append(firstListOfChords[el])
    else:
        if 'continue' in str(firstListOfChords[el].tie):
            if firstListOfChords[el].pitches == firstListOfChords[el-1].pitches:
                pass
            else:
                newListOfChords.append(firstListOfChords[el])
        else:
            newListOfChords.append(firstListOfChords[el])
listOfTransposedChords = []
for el in newListOfChords:
    rootOfChord = el.bass()
    middleC = note.Note('c4')
    rootNote = note.Note(rootOfChord)
```

307	transpositionInterval = interval.notesToChromatic(rootNote, middleC)	
308	transpositionIntervalSemitones = transpositionInterval.semitones	
309	transposedChord = el.transpose(transpositionIntervalSemitones)	
310	listOfTransposedChords.append(transposedChord)	
311	listOfTransposedChordSequences = []	
312	listOfPositions = list(range(0, len(listOfTransposedChords)))	
313	for el in listOfPositions:	
314	if el < len(listOfTransposedChords)-1:	
315	newChordSequence = []	
316	newChordSequence.append(listOfTransposedChords[el])	
317	newChordSequence.append(listOfTransposedChords[el+1])	
318	listOfTransposedChordSequences.append(newChordSequence)	
319	listOfChordSequencePitches = []	
320	for el in listOfTransposedChordSequences:	
321	i = el[0].pitches, el[1].pitches	
322	listOfChordSequencePitches.append(i)	
323	listOfDurations = []	
324	results3 = Counter()	
325	for el in listOfChordSequencePitches:	
326	results3[el] = listOfChordSequencePitches.count(el)	
327	listOfSummedDurations = []	
328	listOfSummedDurations.append(el)	
329	listOfSummedDurations.append(results3[el])	

```
        listOfDurations.append(listOfSummedDurations)
    tidyListOfDurations = []
    for el in listOfDurations:
        if el in tidyListOfDurations:
            pass
        else:
            tidyListOfDurations.append(el)
    for el in tidyListOfDurations:
        dataCategory = 'Chord Sequence (2 Chords - Chord Types - Pitches)'
        chordName = attempt4(str(el[0]))
        chordLength = 'N/A'
        totalDuration = 'N/A'
        numberOfTimesStruck = str(el[1])
        csvWriter()

for el in pieceName:
    thePiece = converter.parse(el)
    fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")
```

```
chordifiedPiece = thePiece.stripTies().chordify()
chords = chordifiedPiece.flat.recurse().getElementsByClass('Chord')
firstListOfChords = []
for el in chords:
    firstListOfChords.append(el)
firstListOfPositions = list(range(0, len(firstListOfChords)))
newListOfChords = []
for el in firstListOfPositions:
    if 'stop' in str(firstListOfChords[el].tie):
        if firstListOfChords[el].pitches == firstListOfChords[el-1].pitches:
            pass
        else:
            newListOfChords.append(firstListOfChords[el])
    else:
        if 'continue' in str(firstListOfChords[el].tie):
            if firstListOfChords[el].pitches == firstListOfChords[el-1].pitches:
                pass
            else:
                newListOfChords.append(firstListOfChords[el])
        else:
            newListOfChords.append(firstListOfChords[el])
listOfTransposedChords = []
for el in newListOfChords:
```

```
        rootOfChord = el.bass()
        middleC = note.Note('c4')
        rootNote = note.Note(rootOfChord)
        transpositionInterval = interval.notesToChromatic(rootNote, middleC)
        transpositionIntervalSemitones = transpositionInterval.semitones
        transposedChord = el.transpose(transpositionIntervalSemitones)
        listOfTransposedChords.append(transposedChord)
    secondListOfChords = []
    for el in listOfTransposedChords:
        secondListOfChords.append(el.orderedPitchClassesString)
    listOfTransposedChordSequences = []
    listOfPositions = list(range(0, len(secondListOfChords)))
    for el in listOfPositions:
        if el < len(secondListOfChords)-1:
            newChordSequence = []
            newChordSequence.append(secondListOfChords[el])
            newChordSequence.append(secondListOfChords[el+1])
            listOfTransposedChordSequences.append(newChordSequence)
    listOfChordSequencePitches = []
    for el in listOfTransposedChordSequences:
        i = el[0], el[1]
        listOfChordSequencePitches.append(i)
    listOfDurations = []
```

```
results3 = Counter()
for el in listOfChordSequencePitches:
    results3[el] = listOfChordSequencePitches.count(el)
    listOfSummedDurations = []
    listOfSummedDurations.append(el)
    listOfSummedDurations.append(results3[el])
    listOfDurations.append(listOfSummedDurations)
tidyListOfDurations = []
for el in listOfDurations:
    if el in tidyListOfDurations:
        pass
    else:
        tidyListOfDurations.append(el)
temporaryListOfData = []
for el in tidyListOfDurations:
    newChordName = []
    for j in el[0]:
        for i in j:
            if i == '<':
                newChordName.append('(')
            else:
                if i == '>':
                    newChordName.append(')')
```

```
                else:
                    if i == 'A':
                        newChordName.append('10')
                    else:
                        if i == 'B':
                            newChordName.append('11')
                        else:
                            newChordName.append(i)
        newChordData = []
        newChordData.append(newChordName)
        newChordData.append(el[1])
        temporaryListOfData.append(newChordData)
    finalListOfData = []
    for el in temporaryListOfData:
        temporaryFinalData = []
        i = str(el[0]).replace("'", "").replace("[", "").replace("]", "").replace(",", "").replace("( ", "(").replace(" )", ")")
        temporaryFinalData.append(i)
        temporaryFinalData.append(el[1])
        finalListOfData.append(temporaryFinalData)
    for el in finalListOfData:
        dataCategory = 'Chord Sequence (2 Chords - Chord Types - Pitch Classes)'
        chordName = attempt4(str(el[0]))
        chordLength = 'N/A'
```

totalDuration = 'N/A'	
numberOfTimesStruck = str(el[1])	
csvWriter()	
for el in pieceName:	
thePiece = converter.parse(el)	
fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")	
listOfNotes = []	
notes = thePiece.flat.recurse().notes	Creates a list of notes (includes chords) in the piece
listOfNotes = []	
for el in notes:	
listOfNotes.append(el)	
listOfPitchesAndSeconds = []	
for el in listOfNotes:	Creates new sublists, each with a pitch and its duration
newNote = []	
if len(el.pitches) == 1:	
listOfPitchesinSingleNote = []	

457	listOfPitchesinSingleNote.append(el.pitches)	
458	firstRevisedNewList = []	
459	for c in listOfPitchesinSingleNote:	
460	for d in c:	
461	newNote.append(d)	
462	newNote.append(el.seconds)	
463	listOfPitchesAndSeconds.append(newNote)	
464	else:	
465	listOfPitchesInChord = []	
466	listOfPitchesInChord.append(el.pitches)	
467	revisedNewList = []	
468	for h in listOfPitchesInChord:	
469	for i in h:	
470	newNote.append(i)	
471	newNote.append(el.seconds)	
472	listOfPitchesAndSeconds.append(newNote)	
473	listOfPitches = []	
474	for el in listOfPitchesAndSeconds:	
475	listOfPitches.append(el[0])	
476	setOfPitches = []	
477	totalListOfPitches = []	Creates a list of all pitches
478	for el in listOfPitchesAndSeconds:	
479	totalListOfPitches.append(el[0])	

firstCleanList = []	
for el in totalListOfPitches:	Creates a list of all different pitches used in the piece
if str(el) in firstCleanList:	
pass	
else:	
firstCleanList.append(str(el))	
finalListOfDurations = []	
for el in firstCleanList:	Creates sublists of each different pitch used in the piece and the durations of each statement
initialListOfSeconds = []	
for i in listOfPitchesAndSeconds:	
if el == str(i[0]):	
initialListOfSeconds.append(i[1])	
listOfSummedDurations = []	
listOfSummedDurations.append(el)	
listOfSummedDurations.append(sum(initialListOfSeconds))	
finalListOfDurations.append(listOfSummedDurations)	
newListOfNotes = []	
for el in notes:	Creates a new list of each pitch used in the piece, note including those that are tied to
if 'stop' in str(el.tie):	
pass	
else:	
if 'continue' in str(el.tie):	
pass	

`        else:`	
`            newListOfNotes.append(el)`	
`    newNewListOfPitches = []`	
`    listOfPitches = []`	
`    for el in newListOfNotes:`	
`        newNewListOfPitches.extend(el.pitches)`	
`    listOfPitchClasses = []`	
`    for el in newNewListOfPitches:`	
`        listOfPitchClasses.append(str(el))`	
`    draftListOfPitches = []`	
`    results3 = Counter()`	Counts each pitch struck in the piece
`    for el in listOfPitchClasses:`	
`        results3[el] = listOfPitchClasses.count(el)`	
`        countedListOfPitches = []`	
`        countedListOfPitches.append(el)`	
`        countedListOfPitches.append(results3[el])`	
`        draftListOfPitches.append(countedListOfPitches)`	
`    tidyListOfPitches = []`	
`    for el in draftListOfPitches:`	
`        if el in tidyListOfPitches:`	
`            pass`	
`        else:`	
`            tidyListOfPitches.append(el)`	

finalTotalListOfBoth = []	
for el in finalListOfDurations:	Creates sublists of each different pitch, its duration, and number of statements
fullData = []	
for p in tidyListOfPitches:	
if p[0] == el[0]:	
fullData.append(el[0])	
fullData.append(el[1])	
fullData.append(p[1])	
finalTotalListOfBoth.append(fullData)	
for el in finalTotalListOfBoth:	Appends the relevant data to the relevant variable, and writes it to the CSV file
dataCategory = 'Pitches'	
chordName = attempt4(str(el[0]))	
totalDuration = el[1]	
chordLength = '1'	
numberOfTimesStruck = str(el[2])	
csvWriter()	
for el in pieceName:	
thePiece = converter.parse(el)	
fileTitle = str(el).replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Drei Kleine Stücke/Complete/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Fünf Stücke für Orchester/Sounding/", "").replace("~/Desktop/Cambridge/MPhil/Dissertation/MPhil Corpus/Sechs Bagatellen für	

	Streichquartett/", "").replace("SOUNDING", "").replace("DRAFT 1", "").replace("FULL", "").replace(".mxl", "")	
546	listOfNotes = []	
547	notes = thePiece.flat.recurse().notes	
548	newListOfNotes = []	
549	for el in notes:	
550	if 'stop' in str(el.tie):	
551	pass	
552	else:	
553	if 'continue' in str(el.tie):	
554	pass	
555	else:	
556	newListOfNotes.append(el)	
557	newNewListOfPitches = []	
558	listOfPitches = []	
559	for el in newListOfNotes:	
560	newNewListOfPitches.extend(el.pitches)	
561	listOfPitchClasses = []	
562	for el in newNewListOfPitches:	
563	listOfPitchClasses.append(el.pitchClass)	
564	listOfPitchClasses.sort()	
565	draftListOfPitches = []	
566	results3 = Counter()	

```
for el in listOfPitchClasses:
    results3[el] = listOfPitchClasses.count(el)
    countedListOfPitches = []
    countedListOfPitches.append(el)
    countedListOfPitches.append(results3[el])
    draftListOfPitches.append(countedListOfPitches)
tidyListOfPitches = []
for el in draftListOfPitches:
    if el in tidyListOfPitches:
        pass
    else:
        tidyListOfPitches.append(el)
listOfNotes = []
for el in notes:
    listOfNotes.append(el)
listOfPitchesAndSeconds = []
for el in listOfNotes:
    newNote = []
    if len(el.pitches) == 1:
        listOfPitchesinSingleNote = []
        listOfPitchesinSingleNote.append(el.pitches)
        firstRevisedNewList = []
        for c in listOfPitchesinSingleNote:
```

590	for d in c:	
591	newNote.append(d.pitchClass)	
592	newNote.append(el.seconds)	
593	listOfPitchesAndSeconds.append(newNote)	
594	else:	
595	listOfPitchesInChord = []	
596	listOfPitchesInChord.append(el.pitches)	
597	revisedNewList = []	
598	for h in listOfPitchesInChord:	
599	for i in h:	
600	newNote.append(i.pitchClass)	
601	newNote.append(el.seconds)	
602	listOfPitchesAndSeconds.append(newNote)	
603	listOfPitchesAndSeconds.append(newNote)	
604	listOfPitches = []	
605	for el in listOfPitchesAndSeconds:	
606	listOfPitches.append(el[0])	
607	setOfPitches = set(listOfPitches)	
608	finalListOfDurations = []	
609	for el in setOfPitches:	
610	initialListOfSeconds = []	
611	for i in listOfPitchesAndSeconds:	
612	if el == i[0]:	

```
            initialListOfSeconds.append(i[1])
        listOfSummedDurations = []
        listOfSummedDurations.append(el)
        listOfSummedDurations.append(sum(initialListOfSeconds))
        finalListOfDurations.append(listOfSummedDurations)
    finalTotalListOfBoth = []
    for el in finalListOfDurations:
        fullData = []
        for p in tidyListOfPitches:
            if p[0] == el[0]:
                fullData.append(el[0])
                fullData.append(el[1])
                fullData.append(p[1])
                finalTotalListOfBoth.append(fullData)
    for el in finalTotalListOfBoth:
        dataCategory = 'Pitch Classes'
        chordName = el[0]
        totalDuration = el[1]
        chordLength = '1'
        numberOfTimesStruck = el[2]
        csvWriter()
```

Appendix 2

The full results of this study are not given here simply due to economy of space: they comprise over 6000 lines of data, which would require over 100 pages to display. In addition, they are extremely unwieldy in paper form. Nonetheless, in the interests of transparency, and to aid possible future research, they are accessible as an Excel document at the following URL:
https://www.joshuaballance.co.uk/writing