ANALYZING PITCH ESTIMATION ACCURACY IN CROSS-TALK SCENARIOS: A STUDY WITH WIND INSTRUMENTS

Peter MEIER (peter.meier@audiolabs-erlangen.de) (0000-0002-3094-1931)[1],
Meinard MÜLLER (meinard.mueller@audiolabs-erlangen.de) (0000-0001-6062-7524)[1], and
Stefan BALKE (stefan.balke@audiolabs-erlangen.de) (0000-0003-1306-3548)[1]

[1]**International Audio Laboratories Erlangen**, Am Wolfsmantel 33, Erlangen, 91058 Germany

ABSTRACT

Intonation accuracy is crucial for wind instrument ensembles, where pitch deviations affect harmonic coherence. Music Information Retrieval (MIR) techniques, particularly pitch estimation, offer potential for real-time intonation monitoring. However, in natural ensemble settings, microphone cross-talk can compromise pitch accuracy. In this article, we systematically investigate the impact of cross-talk on pitch estimation for wind instruments using the `ChoraleBricks` dataset, which contains multitrack recordings of isolated choral performances. By simulating cross-talk scenarios with Gaussian noise, single- and multi-instrument interference, we assess the robustness of lightweight, real-time capable estimators like `YIN` and `SWIPE` against more advanced methods like `PYIN` and `CREPE`. Our results show that pitch estimation accuracy declines significantly below an SNR threshold of 15 dB. To address this, we identify instrument-specific challenges and propose frequency filtering to mitigate cross-talk interference. These findings inform the development of robust, real-time intonation monitoring systems for wind ensembles, with applications in music education, performance analysis, and rehearsal optimization.

1. INTRODUCTION

Cross-talk is a common challenge in ensemble music recording, occurring when a microphone captures not only its intended instrument but also unintended sounds from other sources, as illustrated in Figure 1. This interference introduces artifacts in recorded audio, significantly affecting tasks such as pitch estimation and intonation monitoring. Previous research has explored various strategies to mitigate cross-talk. Prätzlich et al. [1] proposed methods to reduce interference in multi-channel recordings, while Seipel and Lerch [2] simulated cross-talk using anechoic orchestral recordings to address multi-track recording challenges. Maher and Beauchamp [3] examined pitch estimation in noisy environments, focusing on instrument-specific noise such as flute turbulence. Similarly, Singh et al. [4] assessed the noise robustness of pitch estimation, using their *DeepF0* method to evaluate accuracy under varying levels of accompaniment noise.

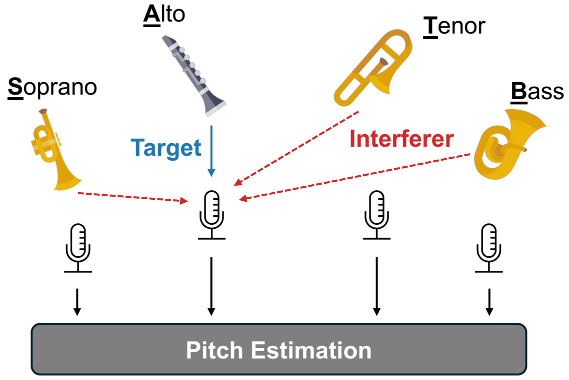

Figure 1. Cross-talk in wind instrument quartets: Each microphone records its designated instrument's sound (target) and unintended signals from other instruments (interferer).

In this work, we investigate pitch estimation accuracy in cross-talk scenarios, specifically for wind instruments, with applications in intonation monitoring in mind. We evaluate traditional methods such as `YIN` [5] and `SWIPE` [6], along with more advanced techniques like `PYIN` [7] and `CREPE` [8]. Our experiments reveal that `YIN` and `SWIPE` achieve a good balance of robustness, efficiency, and real-time capability for this application.

The main contributions of this article are as follows. First, we systematically investigate cross-talk for various pitch estimation approaches, revealing its significant impact on accuracy and establishing a 15 dB SNR threshold for reliable detection. Second, we leverage a novel dataset of 13 wind instruments to enable systematic experiments in cross-talk scenarios — a setting rarely explored in MIR research. Third, we provide insights into instrument-specific challenges and propose practical solutions, such as frequency filtering, to improve algorithmic robustness. Although these topics are not entirely new, this study highlights wind instruments as an important research area and is a step toward creating tools that benefit wind instrument players in real-world applications.

The paper is organized as follows: Section 2 describes our application scenario and dataset. Section 3 presents baseline results, while Section 4 explores three interference types: Gaussian noise, single-instrument, and multi-instrument cross-talk. Finally, Section 5 summarizes our findings and suggests future directions. Audio examples

ISBN 978-3-200-10642-0

and additional materials are available on a dedicated website [1].

2. APPLICATION AND DATASET

2.1 Application

In wind instrument ensembles, real-time pitch and intonation monitoring systems could provide valuable support during rehearsals, recordings, and live concerts. By analyzing multiple microphone channels simultaneously, these systems can offer conductors and musicians not only insight into individual instrument intonation but also into ensemble tuning and any pitch drift during performances. These tools can help quickly resolve intonation issues by identifying the specific instruments that need adjustment. While real-time pitch monitoring tools, such as the Python application Pytch, [2] are already utilized in singing applications, cross-talk is usually mitigated using special larynx microphones. With the microphone setups typically used for wind instruments, cross-talk on individual instrument microphones is a potential issue that we aim to investigate through our experiments.

2.2 Dataset

For our experiments, we use the `ChoraleBricks` dataset [9] to simulate the cross-talk scenario. This dataset includes multi-track recordings of ten four-part chorales, with each part, soprano (`S`), alto (`A`), tenor (`T`), and bass (`B`), recorded separately but in sync using a conducting video. Table 1 provides an overview of the wind instruments in the dataset. For detailed track distribution across the chorales, refer to [9]. This dataset is ideal for comparing pitch estimation among different instrument tracks and allows for creating mixes with diverse instrument combinations. For instance, one ensemble might consist solely of brass instruments like two trumpets, a baritone, and a tuba, while another mix could combine brass and woodwinds like trumpet, clarinet, tenor saxophone, and tuba.

Besides the multi-track recordings, `ChoraleBricks` includes F0 annotations, generated interactively using *Sonic Visualiser* (v5.0.1) [10] and the *pYIN VAMP plugin* (v3) [7]. These annotations were verified using sonification methods [11] and manually corrected if necessary. For the tuba (`tba`), a salience-based F0 estimator was used.

Figure 2 shows the MIDI pitch distribution for each instrument in the dataset, sorted by their annotated F0 median frequency. This order will be maintained in later analyses. The instruments roughly fall into two groups based on pitch distribution: Group 1, from oboe (`ob`) to English horn (`eh`), and Group 2, from French horn (`fho`) to trombone (`tb`). Flutes (`fl`) play an octave higher than Group 1, reaching the dataset's maximum frequency of 1250.97 Hz, while tubas (`tba`) play an octave lower than Group 2, with a minimum frequency of 38.19 Hz. Some instruments, such as the alto saxophone (`as`), trumpet (`tp`), flugelhorn (`fh`), and clarinet (`cl`), share a similar pitch distribution, each with 20 recorded tracks. Others, like the English horn (`eh`) and French horn (`fho`), have a more restricted range with 10 or 11 tracks, respectively (see Table 1).

ID	Instrument	Type	Part	#
`fl`	Flute	`W`	`S`	10
`ob`	Oboe	`W`	`S`	10
`as`	Alto Saxophone	`W`	`SA`	20
`tp`	Trumpet	`Br`	`SA`	20
`fh`	Flugelhorn	`Br`	`SA`	20
`cl`	Clarinet	`W`	`SA`	20
`eh`	English Horn	`W`	`A`	10
`fho`	French Horn	`Br`	`AT`	11
`bar`	Baritone	`Br`	`SATB`	24
`bs`	Baritone Saxophone	`W`	`TB`	18
`bcl`	Bass Clarinet	`W`	`TB`	12
`tb`	Trombone	`Br`	`TB`	8
`tba`	Tuba	`Br`	`B`	10

Table 1. IDs, types, parts, and track counts per instrument in the `ChoraleBricks` dataset. Instrument families: `W` (woodwind), `Br` (brass). Part categories: `S` (soprano), `A` (alto), `T` (tenor), `B` (bass).

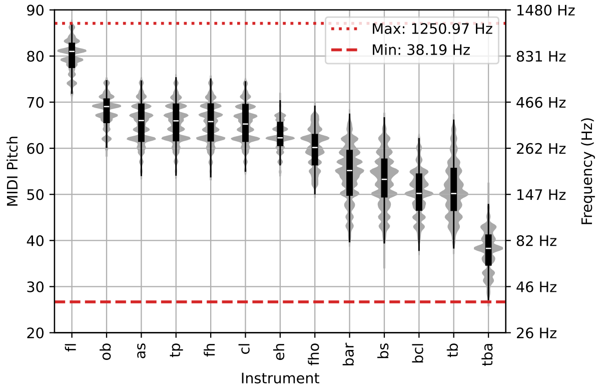

Figure 2. MIDI pitch distribution of `ChoraleBricks` by instrument, based on F0 reference annotations, sorted by median frequency (white markers). Red dashed lines show the dataset's minimum and maximum frequencies.

3. BASELINE EXPERIMENTS

In this section, we present baseline experiments that completely eliminate cross-talk, as all instruments in the `ChoraleBricks` dataset were recorded in isolation. These experiments serve as reference points and establish an upper bound for evaluating the impact of cross-talk on pitch estimation in Section 4. We first introduce the pitch estimation algorithms (Section 3.1), followed by the evaluation metrics (Section 3.2), and then analyze pitch accuracy for different instruments (Section 3.3) and MIDI pitches (Section 3.4).

3.1 Pitch Estimators

For our experiments, we use four widely adopted pitch estimation algorithms, each with distinct characteristics.

[1] https://www.audiolabs-erlangen.de/resources/MIR/2025-SMC-PitchCrosstalk

[2] https://github.com/pytchtracking/pytch

ISBN 978-3-200-10642-0

RPA		CREPE			PYIN			YIN			SWIPE		
Cent Tol.		10	25	50	10	25	50	10	25	50	10	25	50
Woodwind	fl	0.907	0.971	0.979	0.938	0.971	0.984	0.939	0.965	0.976	0.776	0.953	0.966
	ob	0.969	0.989	0.993	0.954	0.975	0.983	0.952	0.969	0.976	0.962	0.986	0.993
	as	0.946	0.974	0.986	0.938	0.975	0.990	0.939	0.974	0.988	0.900	0.967	0.985
	cl	0.963	0.987	0.993	0.943	0.970	0.979	0.945	0.970	0.979	0.953	0.985	0.992
	eh	0.969	0.990	0.993	0.950	0.979	0.986	0.942	0.967	0.974	0.966	0.990	0.995
	bs	0.945	0.979	0.987	0.939	0.975	0.989	0.938	0.972	0.985	0.932	0.977	0.990
	bcl	0.897	0.946	0.973	0.946	0.968	0.977	0.950	0.970	0.980	0.958	0.981	0.988
	∅	0.942	0.977	0.986	0.944	0.973	0.984	0.943	0.970	0.980	0.921	0.977	0.987
Brass	tp	0.964	0.987	0.994	0.933	0.971	0.981	0.937	0.970	0.981	0.943	0.985	0.995
	fh	0.959	0.984	0.993	0.929	0.969	0.982	0.933	0.968	0.981	0.910	0.983	0.993
	fho	0.940	0.978	0.990	0.924	0.971	0.988	0.923	0.966	0.982	0.912	0.983	0.995
	bar	0.895	0.933	0.947	0.873	0.922	0.938	0.876	0.918	0.933	0.817	0.931	0.952
	tb	0.917	0.967	0.975	0.920	0.974	0.983	0.926	0.973	0.982	0.897	0.962	0.972
	tba	0.700	0.867	0.911	0.719	0.877	0.940	0.716	0.876	0.937	0.713	0.878	0.945
	∅	0.896	0.953	0.968	0.883	0.947	0.969	0.885	0.945	0.966	0.865	0.954	0.975
∅ **Overall**		0.928	0.967	0.979	0.918	0.961	0.976	0.919	0.959	0.973	0.898	0.967	0.982

Table 2. Mean RPA values for different estimators (CREPE, PYIN, YIN, SWIPE) using different tolerance values in cents (10, 25, 50) for woodwind and brass instruments, sorted by median frequencies.

First, YIN [5], an autocorrelation-based method, is known for its simplicity and speed but is prone to octave errors. Second, SWIPE [6] leverages a sawtooth waveform model to match spectral templates, reducing subharmonic errors by emphasizing fundamental and prime harmonics. This approach can be adapted for real-time applications with modifications [12]. Third, PYIN [7], an extension of YIN, incorporates a hidden Markov model to improve pitch estimates by considering temporal context, though it is not suited for real-time use. Lastly, CREPE [8], a deep learning-based method using a convolutional neural network, achieves state-of-the-art accuracy but is computationally intensive.

For YIN and PYIN, we use the implementations from librosa[3] [13]. For SWIPE, we rely on the libf0 Python package [14]. For CREPE, we use the official implementation available on GitHub.[4] This version of CREPE is trained on more data than reported in the original paper and includes an improved argmax-local weighted averaging formula, enhancing accuracy compared to the original publication.

In our experiments, we use a sample rate of 44100 Hz, a hop size of 512 (≈ 11.61 ms), and a window size of 4096 (≈ 92.88 ms). For CREPE, we maintain the default step size of 10 ms (hop size of 441) for better accuracy. We do not apply the CREPE Viterbi post-processing for temporal smoothing as it is not part of the original model paper. The frequency range is set between note C1 (≈ 32.70 Hz) and note A6 (1760 Hz) to cover our dataset's full range (see Figure 2). The pitch resolution is set to 10 cents for SWIPE and PYIN. Implementations of YIN and CREPE lack this parameter, but their post-processing methods, such as parabolic interpolation and local averaging, result in non-quantized frequency resolution.

3.2 Evaluation Metrics

In our experiments, we use evaluation metrics from the mir_eval Python package [15], focusing on Raw Pitch Accuracy (RPA) [16], which measures the percentage of melody frames where the estimated frequency matches the reference within a specified cent tolerance. To assess its impact on pitch estimation accuracy, we evaluate three tolerances: 50 cents (half a semitone), 25 cents, and 10 cents. While 50 cents is commonly used in pitch estimation evaluation, it may be too broad for applications like intonation analysis. Smaller tolerances, such as 25 and 10 cents, impose stricter requirements and pose greater challenges for pitch estimation algorithms. We do not include Voicing Recall (VR) in this study, as our primary focus is on pitch accuracy in cross-talk scenarios.

3.3 Analysis Across Instruments

In Table 2, we present an overview of the mean RPA values for different pitch estimators, using various evaluation tolerances, shown for individual instruments. Additionally, we provide the average RPA values for the two subsets of instrument types, brass and woodwind, along with the overall average for the entire dataset.

Analyzing the overall RPA at a tolerance of 50 cents, all estimators perform comparably well, with SWIPE achieving the highest RPA value of 0.982. When the tolerance is reduced to 25 cents, accuracy declines slightly, with RPA values ranging from 0.959 for YIN to 0.967 for SWIPE and CREPE. At a lower tolerance of 10 cents, accuracy decreases more noticeably, with values spanning from 0.898 for SWIPE to 0.928 for CREPE.

[3] We specifically used commit ebd878f, which includes recent updates and bug fixes for YIN and PYIN.

[4] https://github.com/marl/crepe

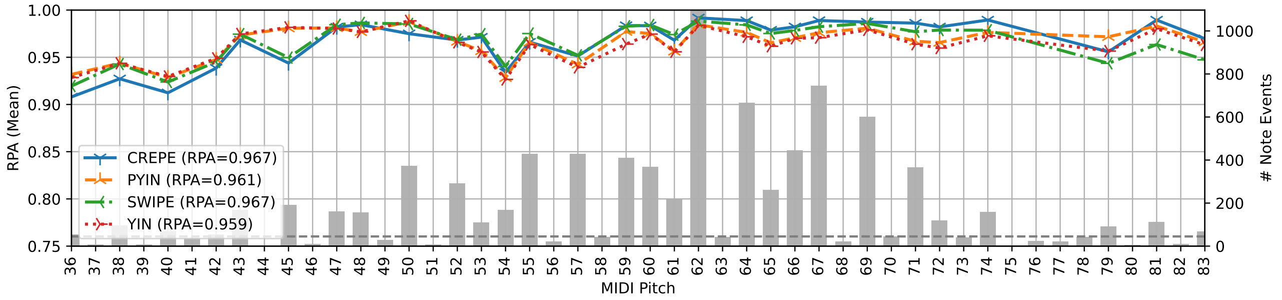

Figure 3. Pitch-dependent evaluation of pitch estimation methods on the `ChoraleBricks` dataset, displaying mean RPA values for estimators with a 25-cent tolerance. Gray bars show the number of note events per MIDI pitch from annotations. RPA data points are plotted only for notes with more than 45 events (gray dashed line).

In the following, we examine the RPA values with a 10-cent tolerance to highlight key differences and challenges among pitch estimation algorithms and instruments. On average, woodwind instruments are detected more accurately than brass instruments. For woodwinds, `PYIN` achieves the highest RPA at 0.944, while for brass instruments, `CREPE` leads with an RPA of 0.896. The tuba (`tba`), the most challenging instrument in the dataset, performs below its group's average and has its best RPA at 0.719 with `PYIN`, whereas `CREPE` yields 0.700. Although the flute (`fl`) is the highest-pitched instrument in the dataset, it shows weaker results: from 0.776 with `SWIPE` to 0.939 with `YIN`, possibly due to its turbulent noise or breathiness [3].

In this analysis, `CREPE` generally emerges as the top-performing method. However, traditional methods like `YIN` or `SWIPE` sometimes slightly outperform `CREPE`, such as with the tuba (`tba`) or even exceed its performance, as seen with the bass clarinet (`bcl`). This may be because certain instruments are not part of `CREPE`'s training set, leading to less effective generalization. Incorporating wind instruments into `CREPE`'s training set could enhance its effectiveness and is a potential direction for future research.

While 10-cent tolerances are desirable for our application, they can lead to problems for three key reasons. First, many methods use interpolation techniques like parabolic interpolation or local averaging to achieve the desired output resolution. Second, small implementation details have a larger impact at a 10-cent error tolerance. Third, reference annotations and methods can introduce bias at the 10-cent precision level. Therefore, we will use a 25-cent tolerance for our analyses moving forward, which makes these issues less significant. Internal experiments also showed similar trends between the 10-cent and 25-cent evaluation tolerances.

3.4 Analysis Across MIDI Pitches

Figure 3 shows the pitch-dependent evaluation results of four pitch estimation methods on the `ChoraleBricks` dataset. The figure plots the mean RPA values with a 25-cent tolerance against MIDI pitch values. We also include the number of note events for each MIDI pitch, derived from the F0 annotations. RPA data points are only plotted for MIDI pitches with more than 45 note events, as pitches with fewer events may lead to less reliable RPA values due to insufficient data.

All pitch estimators perform well at the 25-cent tolerance, with mean RPA values mostly exceeding 0.95 and none below 0.9. The pitch curves generally remain flat, although a slight RPA drop is observed for all estimators at MIDI pitches 54 and 57. Below MIDI pitch 43, RPA tends to decrease for all estimators, with `CREPE` showing the lowest accuracy; however, this range exclusively includes notes played by the tuba (`tba`). Above MIDI pitch 74, `SWIPE` shows lower RPA values compared to other methods. This higher range features only a single instrument, the flute (`fl`). The higher and lower MIDI pitches are not supported by as many note events; therefore, they should be interpreted with caution. Additionally, as discussed in Section 3.3, both the flute and tuba have been identified as among the most challenging instruments in this dataset.

Using a 25-cent evaluation tolerance, `CREPE` and `SWIPE` both show the best results with an RPA of 0.967 each. However, this analysis indicates that the RPA of estimators can vary depending on the pitch being analyzed. Specifically, we observe that `CREPE` performs worse for lower pitches, while `SWIPE` struggles more with higher pitches.

4. CROSS-TALK EXPERIMENTS

In this section, we simulate various cross-talk scenarios with increasing complexity: noise interference (Section 4.1), single-instrument interference (Section 4.2), and multi-instrument interference (Section 4.3).

In all three scenarios, we employ a similar signal processing flow to mix target instrument tracks with interfering signals, as illustrated in Figure 4. To clarify the general concept, we use the first scenario as an example (see Section 4.1). First, the target instrument is normalized to -23 LUFS (Loudness Units Full Scale), and a limiter is set at -1 dBFS (decibels relative to full scale) to prevent potential distortion from audio peaks. This setup complies with the *EBU R 128* broadcast standard, ensuring consistent loudness levels across different instrument recordings.

The interferer, whether it is noise, a single instrument, or multiple instruments, is normalized similarly to the tar-

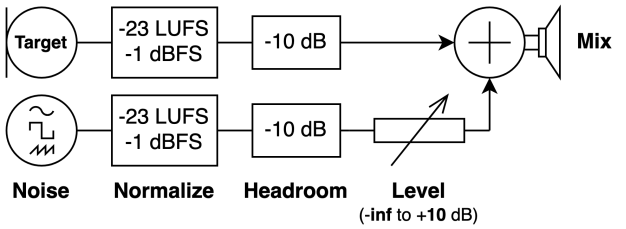

Figure 4. Flowchart for mixing instrument tracks with, e.g., Gaussian noise as interferer. See text for details.

get. The level of the interferer can be adjusted from $-\infty$ (meaning no interference) to +10 dB (where the interferer is even louder than the target). This adjustment allows for SNR levels ranging from ∞ to -10 dB in increments of 5 dB. To prevent audio clipping in the final mix due to the additional 10 dB boost from the interferer, we apply a headroom of -10 dB to both the target and interferer. This ensures that the final mix consistently remains below 0 dBFS for all SNR variations. An SNR level of 0 dB results in equal loudness for both the target and interferer.

4.1 Gaussian Noise Interference

For the first experiment, we add Gaussian noise at various SNR levels to all tracks in the ChoraleBricks dataset, as shown in Figure 5a. Figure 5b presents the mean RPA for different SNR levels and for several pitch estimators. All estimators yield similar results at SNR levels of 15 dB and higher, where accuracy plateaus and further noise reduction is unlikely to enhance RPA results. Below 15 dB, the estimators substantially diverge. For example, CREPE performs best with an RPA of 0.929 at 0 dB SNR, decreasing to 0.760 at -10 dB SNR. SWIPE follows with an RPA of 0.843 at 0 dB SNR, dropping to 0.515 at -10 dB SNR. YIN and PYIN show similar values of 0.732 and 0.713 at 0 dB but experience a significant drop in RPA at lower SNR levels.

Figure 5c illustrates the noise robustness of individual instruments with a 25-cent tolerance, using SWIPE as an example. At an SNR of 15 dB, all instruments achieve RPA values of 0.9 or higher, except for the tuba (tba), which remains below this value (see Table 2). The bass clarinet (bcl) exhibits the highest robustness, maintaining an RPA of 0.72 at -10 dB SNR, closely followed by the tuba (tba) and baritone saxophone (bs). In contrast, the oboe (ob) and English horn (eh) are more sensitive to noise, with RPA dropping to 0.35 and 0.38, respectively, at -10 dB SNR. The flute (fl) is most affected, with its RPA decreasing sharply from 0.49 at 0 dB to 0.13 at -10 dB.

4.2 Single-Instrument Interference

In our second experiment, we introduce an additional instrument to interfere with the target, as shown in Figure 6a. We calculate various combinations of target and interfering instruments for different SNR levels, similar to the noise experiment in Section 4.1. For the following discussion, we use a notation where an instrument is appended with its part; for example, fl-S indicates a flute playing the soprano part, and fho-T denotes a French horn playing the tenor part. For each choral piece, we consider all possible pairs of available instrument tracks, from tba-B to fl-S. We also allow mixing instruments of the same part, like two soprano instruments playing in unison (fl-S, ob-S), and scenarios where a target instrument track is mixed with its own recording, such as (cl-A, cl-A), which serves as a baseline for no interference. This method generates a total of 44,916 musically meaningful pairs of instrument tracks with varying configurations of target instruments, interfering instruments, and SNR levels.

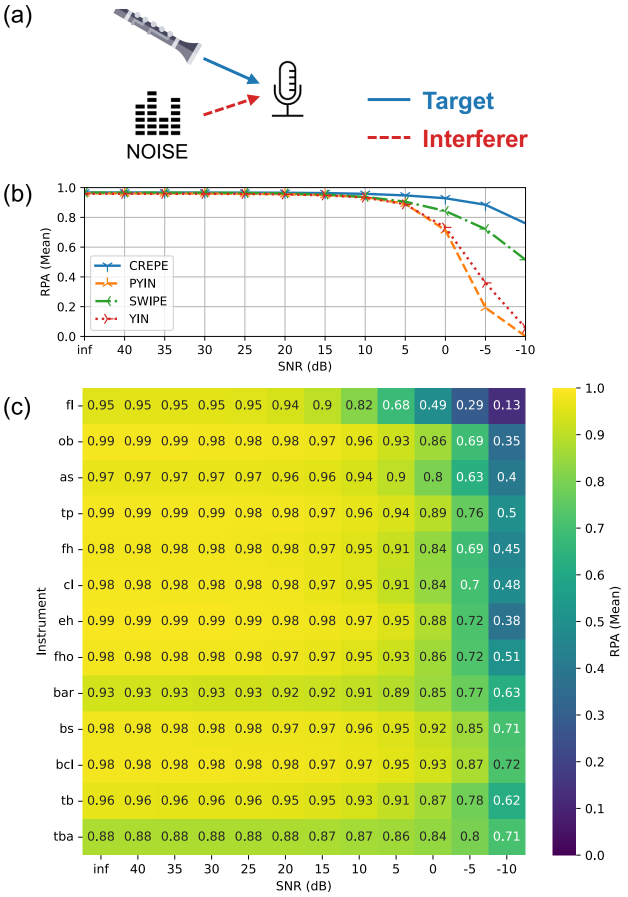

Figure 5. (a) First cross-talk experiment with Gaussian noise. (b) RPA vs. SNR for various pitch estimators at a 25-cent tolerance. (c) RPA vs. SNR for individual instruments using SWIPE.

In Figure 6b, we present the aggregated results of RPA versus SNR for different pitch estimators at a 25-cent tolerance. Estimating the pitch of a target instrument mixed with another interfering instrument is notably more challenging than in the previous noise experiment. At an SNR of 25 dB or lower, all estimators exhibit a substantial decline in accuracy. Even CREPE, which performed well with the noise interferer (RPA of 0.929 at 0 dB), drops to an RPA of 0.571 at 0 dB for the single-instrument interferer. SWIPE, YIN, and PYIN achieve RPA values of 0.507, 0.367, and 0.359 at 0 dB, respectively.

To better understand the conditions under which an estimator prefers one instrument over another, we select a

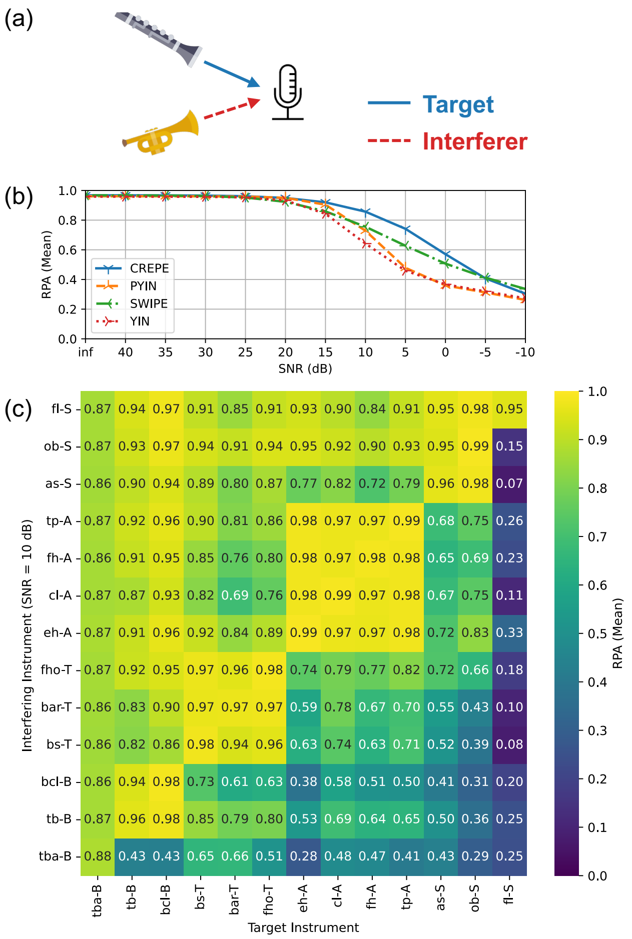

Figure 6. (a) Second cross-talk experiment with single-instrument interference. (b) RPA vs. SNR for different pitch estimators with a 25-cent tolerance. (c) Instrument and part combinations at 10 dB SNR using SWIPE.

representative set of instruments playing different parts (SATB) and analyze their mixes in terms of RPA, as shown in Figure 6c. In the matrix, each column represents a target instrument, while each row shows the interfering instrument mixed with the target at an SNR of 10 dB. We use the SWIPE algorithm as our example estimator. Instruments are sorted by their median frequency (see Figure 2). We ensure an equal balance for all instruments, with each part represented at least three times.

The main diagonal of the matrix, running from the bottom left to the top right, contains pairs where the target and interfering instrument recordings are identical, such as (tba-B, tba-B). These elements have no interference and serve as reference points. Along this diagonal, clusters of instrument parts are observed. The most prominent is the alto group spanning from eh-A to tp-A. Mixing instruments of the same parts means all instruments play the same notes in unison, likely resulting in high RPA values. Similar clusters appear for soprano, tenor, and bass, but do not include the tuba (tba) and flute (fl), which play an octave lower or higher, respectively. Below the diagonal towards the lower right corner of the matrix, RPA values decline drastically. This indicates that high-pitched target instruments show reduced RPA when mixed with low-pitched interferers. Conversely, as indicated by the upper left corner, low-pitched targets are less affected by high-pitched interferers.

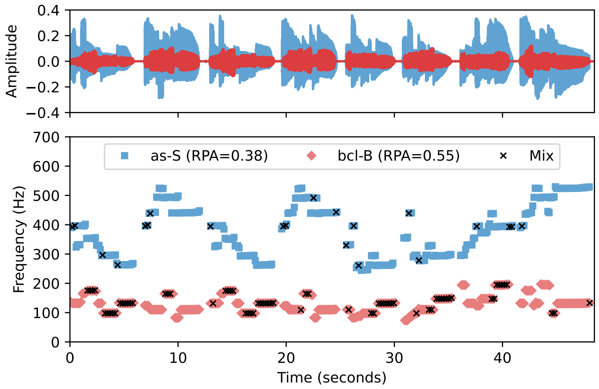

Figure 7. Waveforms and frequency trajectories for "Auf, auf, mein Herz, mit Freuden" by Crüger are shown for alto saxophone (as-S, blue) and bass clarinet (bcl-B, red) at 10 dB SNR. Frequencies are estimated with the SWIPE algorithm and evaluated with a 25-cent tolerance. The estimated trajectory for the mix is in black.

When analyzing various target instrument columns, the flute, which has the highest pitch in our dataset, shows the lowest RPA values and achieves a maximum RPA of only 0.33 when paired with other instruments. In contrast, some lower-pitched instruments, such as the bass clarinet (bcl), often achieve high RPA values above 0.9 in most combinations. An exception occurs when the bass clarinet is mixed with the tuba (tba), which has an even lower pitch; here, the RPA value drops significantly to 0.43.

To explore the impact of low-pitched interferers on high-pitched target instruments, we examine a specific case where the alto saxophone (as-S) is the target, and the bass clarinet (bcl-B) is the interferer. According to Figure 6c, this combination yields an average RPA value of 0.41 across all songs in the dataset. In Figure 7, we analyze a single song ("Auf, auf, mein Herz, mit Freuden" by Crüger), showing waveforms and frequency trajectories for these instruments. Despite a 10 dB level drop for the bass clarinet (bcl) compared to the alto saxophone (as), the alto saxophone achieves an RPA value of 0.38. When switching perspectives and evaluating the mix with the reference annotation of the bass clarinet (bcl), it reaches an RPA value of 0.55. Together, these RPA values total 0.93, indicating that pitch estimation is almost always attributed to one of these instruments. This is illustrated in Figure 7, where the estimated trajectory for the mix (colored in black) consistently aligns with one of the two instrumental trajectories.

This example highlights a significant issue with cross-talk in pitch estimation: F0 algorithms, like SWIPE, are often designed to estimate the lowest pitch in a signal, which becomes challenging when mixed with another instrument playing much lower notes. A straightforward solution may be to use a high-pass filter tailored to the frequency range

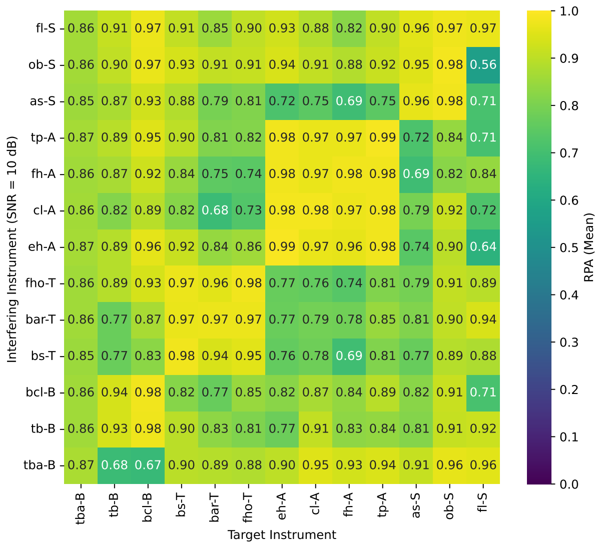

Figure 8. RPA vs. SNR for individual instruments, high-pass filtered at median frequencies, using SWIPE with a 25-cent tolerance.

of the target instrument, as demonstrated in Figure 8. For each mix, we apply a high-pass filter with a cut-off frequency at the median target instrument frequency (see Figure 2) and with a slope of 24 dB per octave. For example, the mix with the oboe (ob) target is high-passed at 441.01 Hz, while the mix with the trombone (tb) target is cut at 148.37 Hz. This significantly reduces the impact of interfering instruments with lower note content compared to the unfiltered experiment (compare Figure 8 with Figure 6c). This approach could be an interesting direction for future research and support real-time intonation monitoring systems.

4.3 Multi-Instrument Interference

In our final experiment, we extend from single-instrument to multi-instrument interference, as shown in Figure 9, simulating typical ensemble performances where instruments are placed close together. We focus on quartets, where each of the four instruments plays a distinct SATB part. Each instrument is analyzed individually as the target while being mixed with a three-instrument interferer at various SNR levels. An SNR of 0 dB indicates equal loudness between the target and interferer, see Section 4 for further details. We limit our experiment to a single piece, selecting the chorale “Auf, auf, mein Herz, mit Freuden” by Crüger, which has the most available instrument tracks. We generate all possible quartet ensembles, ensuring that each instrument plays a distinct SATB part while excluding configurations like SABB or BBBB. This results in 36,000 audio files covering various target instruments, multi-instrument interferers, and SNR-level configurations.

In Figure 9b, we present the RPA values for various SNR levels, aggregated over all mixes for each pitch estimation algorithm and evaluated with a 25-cent tolerance. Compared to the previous experiment with a single interfering instrument, one can observe similar trends for high SNR levels down to 25 dB, where estimators start to diverge. At 15 dB, all estimators maintain an RPA above 0.8. However, for lower SNR levels, the RPA significantly drops, with all estimators falling below 0.5 RPA at an SNR of 0 dB or lower. This suggests that multi-instrument interference with distinct SATB parts poses a greater challenge for pitch estimation than single-instrument interference where some instrument combinations play in unison.

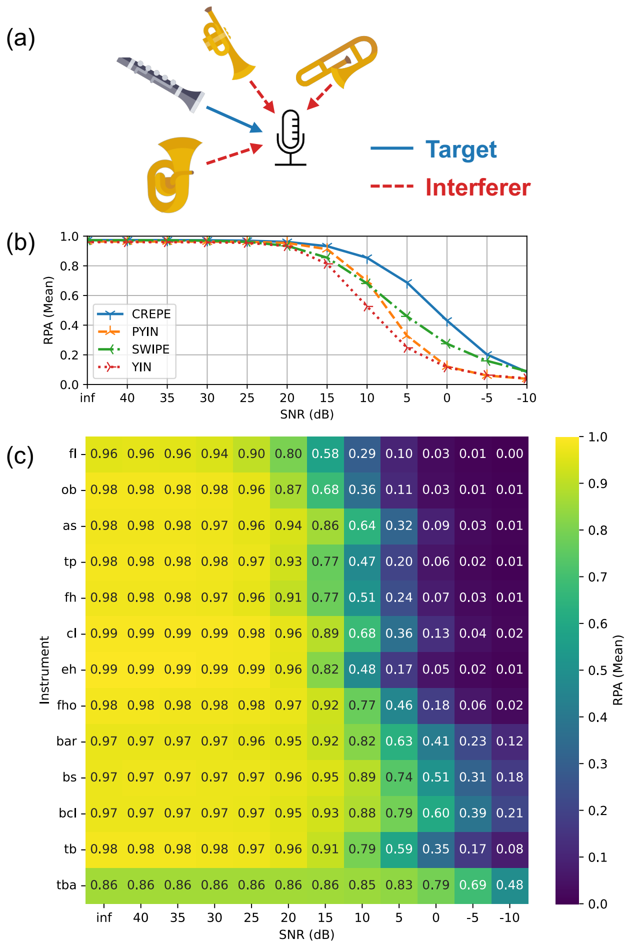

Figure 9. (a) Third cross-talk experiment with multi-instrument interference. (b) RPA vs. SNR for various pitch estimators at 25-cent tolerance. (c) RPA vs. SNR for individual instruments using SWIPE.

Figure 9c illustrates the RPA versus SNR for individual target instruments mixed with a multi-instrument interferer. Unlike the noise experiment in Section 4.1, where most instruments achieved an RPA over 0.9 at 5 dB SNR, achieving similar results in a multi-instrument setting requires a much higher SNR of 20 dB or more. This highlights the increased difficulty for estimators dealing with multi-instrument interference due to increased tonal complexity. Especially high-pitched instruments experience a more rapid decline in RPA with decreasing SNR levels, as they are typically mixed with lower-pitched instruments.

5. CONCLUSIONS

In this study, we examined pitch estimation in wind instrument cross-talk scenarios using the ChoraleBricks dataset. Our results demonstrate the significant impact of cross-talk, particularly at lower SNR levels. We found that YIN and SWIPE offer a strong balance between robustness and real-time capability, while CREPE provides higher accuracy at the cost of increased computational complexity. A key insight from our experiments is that lower-pitched instruments tend to dominate mixed signals, leading to biased pitch estimates. By applying frequency filtering and maintaining an SNR above 15 dB, we can mitigate these effects and improve pitch estimation accuracy.

For future work, we aim to improve pitch estimation models through instrument-specific adaptations, such as enhanced filtering techniques or integrating instrument-specific templates in SWIPE. These adaptations may be complemented by lightweight machine learning techniques trained on data with cross-talk. We also intend to evaluate how cross-talk affects estimation accuracy in real-world recording environments. Finally, we will refine real-time intonation monitoring tools to support practical applications in ensemble performance and music education.

Acknowledgments

The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 500643750 (MU 2686/15-1).

6. REFERENCES

[1] T. Prätzlich, R. Bittner, A. Liutkus, and M. Müller, "Kernel additive modeling for interference reduction in multi-channel music recordings," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015, pp. 584–588.

[2] F. Seipel and A. Lerch, "Multi-Track Crosstalk Reduction Using Spectral Subtraction," in *Proceedings of the Audio Engineering Society (AES) Convention*, Milan, Italy, 2018.

[3] R. C. Maher and J. W. Beauchamp, "Fundamental frequency estimation of musical signals using a two-way mismatch procedure," *The Journal of the Acoustical Society of America*, vol. 95, no. 4, pp. 2254–2263, 1994.

[4] S. Singh, R. Wang, and Y. Qiu, "DeepF0: End-to-end fundamental frequency estimation for music and speech signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Ontario, Canada, 2021, pp. 61–65.

[5] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music." *Journal of the Acoustical Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, 2002.

[6] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[7] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 659–663.

[8] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 161–165.

[9] S. Balke, A. Berndt, and M. Müller, "ChoraleBricks: A modular multitrack dataset for wind music research," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 2025.

[10] C. Cannam, C. Landone, and M. B. Sandler, "Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proceedings of the International Conference on Multimedia*, Florence, Italy, 2010, pp. 1467–1468.

[11] Y. Özer, L. Brütting, S. Schwär, and M. Müller, "libsoni: A Python toolbox for sonifying music annotations and feature representations," *Journal of Open Source Software (JOSS)*, vol. 9, no. 96, pp. 06 524:1–6, 2024.

[12] P. Meier, S. Schwär, G. Krump, and M. Müller, "Evaluating real-time pitch estimation algorithms for creative music game interaction," in *INFORMATIK 2023 – Designing Futures: Zukünfte gestalten*. Bonn, Germany: Gesellschaft für Informatik e.V., 2023, pp. 873–882.

[13] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proceedings the Python Science Conference*, Austin, Texas, USA, 2015, pp. 18–25.

[14] S. Rosenzweig, S. Schwär, and M. Müller, "libf0: A python library for fundamental frequency estimation," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[15] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 367–372.

[16] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.