HIGH ACCURACY AND OCTAVE ERROR IMMUNE PITCH DETECTION ALGORITHMS

M. DZIUBIŃSKI and B. KOSTEK

Multimedia Systems Department
Gdańsk University of Technology
Narutowicza 11/12, 80-952 Gdańsk, Poland
e-mail: kido@sound.eti.pg.gda.pl

The aim of this paper is to present a method improving pitch estimation accuracy, showing high performance for both synthetic harmonic signals and musical instrument sounds. This method employs an Artificial Neural Network of a feed-forward type. In addition, octave error optimized pitch detection algorithm, based on spectral analysis is introduced. The proposed algorithm is very effective for signals with strong harmonic, as well as nearly sinusoidal contents. Experiments were performed on a variety of musical instrument sounds and sample results exemplifying main issues of both engineered algorithms are shown.

1. Introduction

There are two major difficulties, namely, octave errors and pitch estimation accuracy [1–3], that most pitch detection algorithms (PDAs) have to deal with. Octave errors problems, seems to be present in all pitch tracking algorithms, known so far, however, these errors are caused by different input signal properties in the estimation process. In time- domain based algorithms [4–7], i.e., AMDF, modified AMDF [8–10] or normalized cross correlation (NCC) [3, 7, 11], octave errors may be caused by low energy content of odd harmonics. In some cases AMDF or autocorrelation methods are performed first and in addition some information is gathered from calculated spectrum, in order to decrease the possibility of estimation errors [12, 13], resulting in more accurate pitch tracking. Such operations usually require increased computational cost, and larger block sizes, than PDAs working in the time-domain. In the frequency domain, errors are caused mostly by low energy content of the lower order harmonics. In cepstral [2], as well as in autocorrelation of log spectrum (ACOLS) [14] analyses, problems are caused by high energy content in higher frequency parts of the signal. Some algorithms operate directly on time-frequency representation, and are based on analysing trajectories of sinusoidal components in spectrogram (sonogram) of the signal [15, 16]. On the other

hand, estimation accuracy problem for all mentioned domains is caused by a number of samples representing analyzed peaks related to fundamental frequency.

There is an additional problem related to pitch detection. For example, in case of speech signals [1, 17–20], it is very important to determine pitch almost instantaneously, which means that processed frames of the signal must be small. This is because voiced fragments of speech may be very short, with rapidly varying pitch. In case of musical signals, voiced (pitched) fragments are relatively long and pitch fluctuations lower. This property of musical signals enables the use of larger segments of the signal in the pitch estimation procedure. But for both application domains, efficient pitch detection algorithm should estimate pitch periods accurately and smoothly between successive frames, and produce pitch contour that has high resolution in the time-domain.

2. Spectrum peak analysis algorithm

The proposed pitch detection algorithm, a so-called Spectrum Peak Analysis (SPA), is based on analyzing peaks in the frequency domain, representing harmonics of a processed signal. The general concept is based on such relatively easiness of pitch determination by observing signal spectrum and especially intervals between partials that are present in the spectrum. This is independent of the fact that some harmonics may be absent, or they can be partially obscured by the background noise. It should, however, be assumed that they are greater than the energy of the background noise. Estimating pitch contour is performed in block processing, i.e., the signal is divided into blocks with widths depending on pitch estimated for preceding blocks, whereas overlap can be time-varying. The width of the first block is initialized to 4096 samples and is decreased for successive blocks, if the detected pitch is relatively high, and can be represented by lower spectrum resolution. Similarly, if estimated pitch decreases in consecutive blocks, the block width is increased, to provide satisfying spectrum resolution. Each block is weighted by the Hann window.

2.1. Harmonic peak frequency estimation

The first step of the estimation process, performed in each block, is finding one peak that represents any of the signal harmonics. The largest maximum of the spectrum signal is assumed to be one of harmonics, and it is easy to establish its coordinates in terms of frequency. The chosen peak is assumed to be at the M-th harmonic of the signal. In practical experiments $M = 20$ seemed to satisfy all tested sounds, however, setting M to any reasonable value is possible. The natural limitation of this approach is the spectrum resolution. It is assumed that the minimum distance d between peaks representing neighboring harmonics must be four samples. Therefore, if detected maximum index is smaller than $M \cdot d$, M is automatically decreased by the algorithm to satisfy the formulated condition. In some cases, for low frequency signals, block size in the analysis must be suitably large to perform pitch tracking. The next step is calculating M possible

fundamental frequencies, assuming that a chosen harmonic (the largest maximum of the spectrum signal) can be $1, 2, \ldots,$ or M-th harmonic of the analyzed sound:

$$F_{\text{fund}}[i] = \sum_{i=1}^{M} \frac{F_M}{i} \qquad (1)$$

where:

F_{fund} − vector of possible fundamental frequencies,

F_M − frequency of the chosen (largest) harmonic.

The main concept of the engineered algorithm is testing the set of K harmonics related to vector F_{fund}, that are most likely to be peaks representing pitch. The value of K is limited by F_M as follows:

$$K = \text{floor}\left(\frac{F_s}{M}\right) \qquad (2)$$

where:

$\text{floor}(x)$ − returns the largest integer value not greater than x,

F_s − sampling frequency.

Based on M, F_{fund} vector and K, the matrix of frequencies used in analysis can be formed in the following way:

$$FAM(i,j) = \sum_{i=1}^{M} \sum_{j=1}^{K} F_{\text{fund}}[i] \cdot j \qquad (3)$$

where:

FAM − matrix containing frequencies of M harmonics sets.

If M is significantly larger than K, and most energy carrying harmonics are higher order harmonics (the energy of first K harmonics is significantly smaller than, for example, K, $K + 1, \ldots,$ $2 \cdot K$, or higher order harmonics), it is better to choose a set of K consecutive harmonics representing the largest amount of energy. Therefore, frequency of the first harmonic in each set (each row of *FAM*) does not have to represent the fundamental frequency. Starting frequencies of chosen sets can be calculated in the following way:

$$H_{\text{maxset}}[j] = \sum_{i=1}^{K} EH_{(i+j) \cdot F_{\text{fund}}}, \qquad j = 0, ..., L - 1 \qquad (4)$$

where:

H_{maxset} − vector containing energy of consecutive K harmonics for the chosen set, where $H_{\text{maxset}}[k]$ is the sum of K harmonics energies for the following frequencies: $k \cdot F_{\text{fund}}$, $(k + 1) \cdot F_{\text{fund}}, \ldots,$ $(k + K) \cdot F_{\text{fund}}$, EH_{fund} − energy of the harmonic with frequency equal to f, L − dimension of H_{maxset} vector: $L = \text{floor}(\frac{F_s}{F_{\text{fund}}} - K)$.

Starting frequency of each set is based on the index representing the maximum value of H_{maxset}: $F_{\mathrm{start}}[m] = \mathrm{ind}_{\max}[m] \cdot F_{\mathrm{fund}}[m]$ for $m = 1, \ldots, M$.

Finally, modified *FAM* can be formed in the following way:

$$FAM(i, j) = \sum_{i=1}^{M} \sum_{j=1}^{K} F_{\mathrm{start}}[i] + F_{\mathrm{fund}}[i] \cdot (j - 1) \tag{5}$$

2.2. *Harmonic peak analysis*

Each harmonics set, represented by frequencies contained in each row of *FAM* is analyzed in order to evaluate whether it is most likely to be a set of peaks related to fundamental frequency among the remaining $M - 1$ sets. This likelihood is represented by V, while V is calculated for each set in the following way:

$$V = \sum_{i=1}^{K} H_v[i] \tag{6}$$

where:

$H_v[i]$ – value of a spectrum component for i-th frequency for the analyzed set.

If the analyzed spectrum component is not a local maximum – left and right neighboring samples are not smaller than the one assigned to the local maximum, then it is set at 0. Additionally, if local maxima of neighboring regions of spectrum are found, H_v is decreased – values of the maxima found are subtracted from H_v.

Neighboring regions of the spectrum surrounding the frequency F_{H_v}, representing H_v, are limited by the following frequencies:

$$F_L = F_{Hv} - \frac{F_{\mathrm{fund}}}{2} \tag{7a}$$

$$F_R = F_{H_v} + \frac{F_{\mathrm{fund}}}{2} \tag{7b}$$

where:

F_L, F_R – frequency boundaries of spectrum regions surrounding F_{H_v},

F_{fund} – assumed fundamental frequency of the analyzed set.

The fundamental frequency, related to the largest V, is assumed to be the desired pitch of the analyzed signal. As observed from Figs. 1–3, three situations are possible. For example, in Fig. 1, one can see that the analyzed spectrum peak value is not a local maximum, therefore it is set at 0. In addition, local maxima are detected in surrounding regions, which subtracted from H_v give negative values. It is clear that in this situation, it is highly unlikely that H_v is a harmonic. Figure 2 presents a situation in which H_v is a local maximum, and surrounding maxima have small values, opposite to Fig. 3, where analyzed regions contain large local maxima. Therefore Fig. 2 represents a peak that is most likely to be a harmonic.
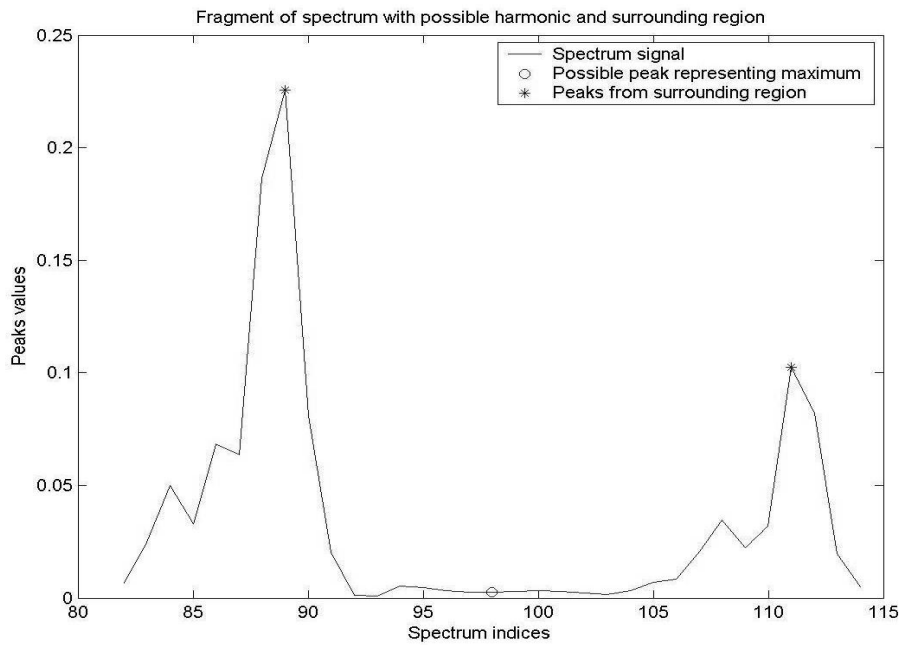
Fig. 1. Analysis of a possible harmonic peak and its surrounding region (analyzed fundamental frequency is not related to peak frequency).
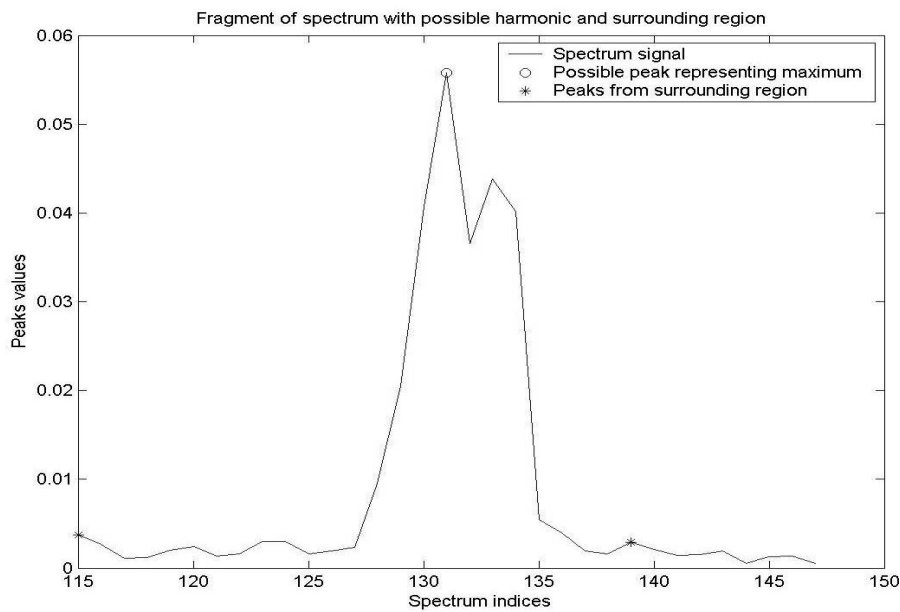

Fig. 2. Analysis of a possible harmonic peak and its surrounding region (analyzed fundamental frequency is correctly related to peak frequency).
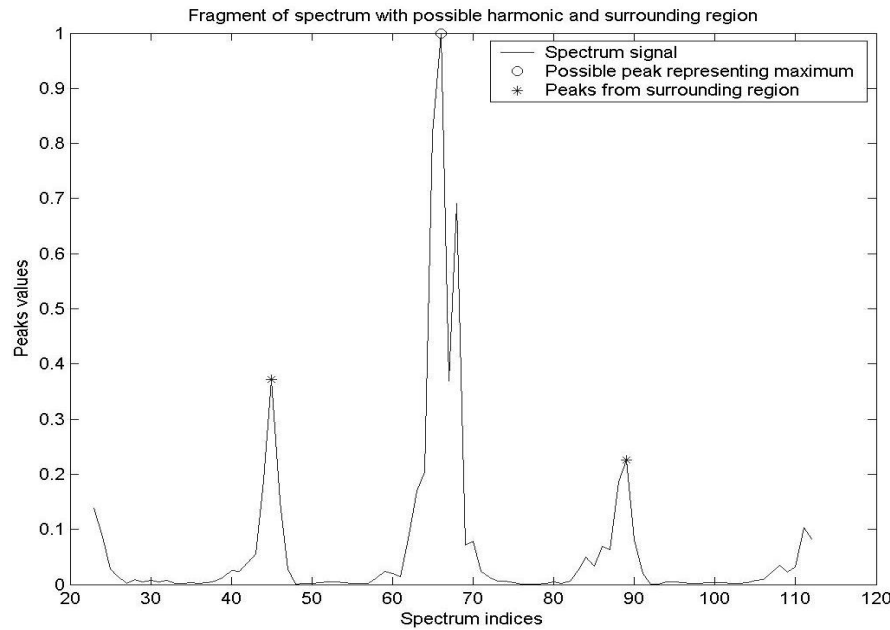
Fig. 3. Analysis of a possible harmonic peak and its surrounding region (analyzed fundamental frequency is two times larger than pitch).

3. Pitch estimation algorithm accuracy

Since the spectrum peak representing pitch is sampled with limited resolution, interpolation is required to improve the algorithm accuracy. Different linear methods have been tested in order to find computationally efficient and suitable interpolation techniques, however, estimating pitch based on a discrete spectrum is not a trivial task. Problems are caused by other frequency components surrounding peak, related to pitch. In practice, those disturbances are caused by spectral leakage of sinusoidal components of a signal (higher order harmonics), and depend on frequency distance between those components and their energy. Therefore, using simple interpolation methods, such as polynomials or splines, would result in a limited performance. Artificial Neural Networks (ANN) seem to be suitable for this task, and are successfully used to improve estimation accuracy, which is shown in the following sections.

3.1. Artificial Neural Network training

Three samples representing spectrum peak related to fundamental frequency have been considered as the ANN input. Index values representing a peak have been normalized to –1, 0 and 1, while 0 was treated as the index of peak maximum and indices –1 and 1 were assumed to be indices of the maximum neighboring samples. Synthetic har-

monic signals were generated to obtain the training input data and target signal. Each training signal was synthesized according to the following formula:

$$S[n] = \sum_{i=1}^{K} \sin\left(\frac{2\pi n i F_{\text{pitch}}}{F_s}\right) \cdot \frac{R[n]}{i} \tag{8}$$

where:

R – vector containing uniformly distributed (on the (0, 1) interval) pseudo-random numbers.

F_{pitch} – fundamental frequency of the synthesized signal,

F_s – sampling frequency,

K – number of harmonics contained in the signal S. K is defined as follows: floor(F_s/F_{pitch}).

It can be observed that a synthetic signal is most likely to have harmonics with decreasing energies, similar to musical instrument sounds. Three training processes were performed, employing various window sizes (different lengths of training signals): 1024, 2048 and 4096 samples, while sampling frequency was equal to 44100. Each signal was weighted by the Hann window, this was because the Hann window was also used in the SPA estimation process. A great number of synthetic signals were generated to obtain training data for each window size, while fundamental frequencies were randomly chosen from F_{min} to 4500 Hz. F_{min} is the lowest possible frequency in respect of d, depending on the window size. The neural network used in the training process was a feed-forward, back-propagation structure with three layers. First layer contained three neurons, the hidden layer – four neurons and the output layer – one neuron. Hyperbolic tangent sigmoid transfer function was chosen to activate the first two layers, whilst the linear identity function was used to activate the last layer. Weights and biases, during the training process, were updated according to Levenberg–Marquardt optimization [21]. Trained network was used in the estimation process, resulting in performance presented in the following section.

3.2. Improved estimation accuracy performance

Pitch estimation accuracy has been tested on synthetic signals, generated according to Eq. 8. Since pitch fluctuations of acoustic sounds can be much greater than the maximum error of the estimation process, using synthetic signals was necessary. The estimation error was calculated in connection with the formulae:

$$f[n] = \frac{(f_{\text{stop}} - f_{\text{start}})(n-1)}{N-1} + f_{\text{start}}, \qquad n = 1, ..., N \tag{9}$$

$$E_{PDA}(f[n]) = \frac{|f[n] - PDA(S_{f[n]})|}{f[n]} 100\% \tag{10}$$

where:

N – number of test frequencies,

f – vector containing test frequencies,

f_{start}, f_{stop} – starting and stopping frequencies of f,

$S_{f[n]}$ – test signal with $f[n]$ pitch.

The proposed SPA algorithm, and in addition, NCC [3] and CA [2] algorithms were implemented in the *Matlab* environment to analyze and compare their performance. Table 1 presents the exemplary average estimation error for the implemented PDAs. Pitch estimations were performed for block size equal to 2048 samples. In addition, improvements of the estimation accuracy for SPA (2-nd order polynomial interpolation and ANN interpolation) are presented, showing the highest performance of the Neural Network-based approach. The average error is understood to be the arithmetic mean of estimation errors calculated in respect of Eq. (10), where $f_{\text{start}} = 50$ Hz, $f_{\text{stop}} = 3000$ Hz and $N = 1000$, while signals had lengths equal to 2048 samples.

Table 1. Average pitch estimation error.

PDA:	NCC	CA	SPA (not optimized)	SPA (polynomial)	SPA (neural network)
Pitch est. error:	3.3399%	2.8518%	0.2808%	0.00122%	0.00000405%

Figures 4–8 presented estimation errors for all tested signals concerning each algorithm, showing error fluctuations over frequency changes. It can be observed that time-domain related algorithms show a decrease in accuracy of estimation when the signal frequency increases, as opposed to frequency-domain related algorithms, where the situation is the opposite.
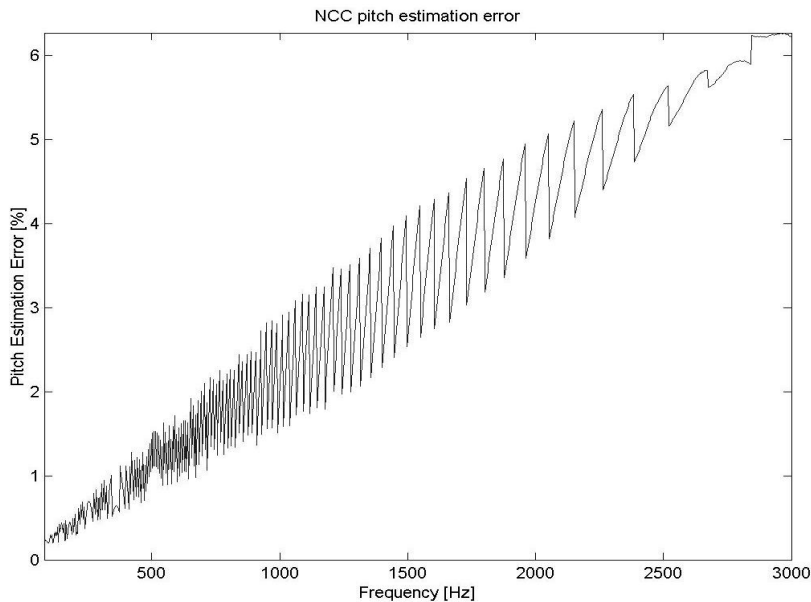

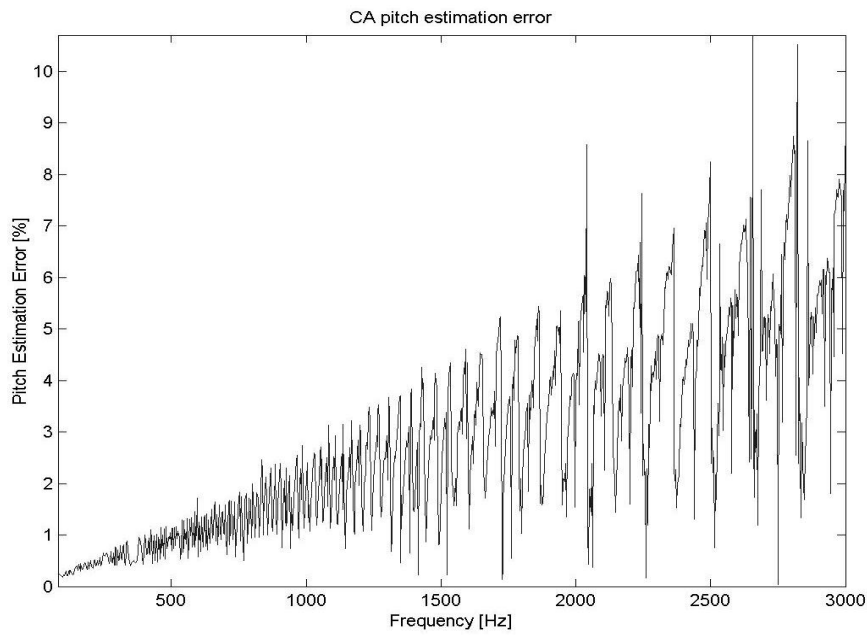Fig. 4. Pitch estimation error of the NCC algorithm.

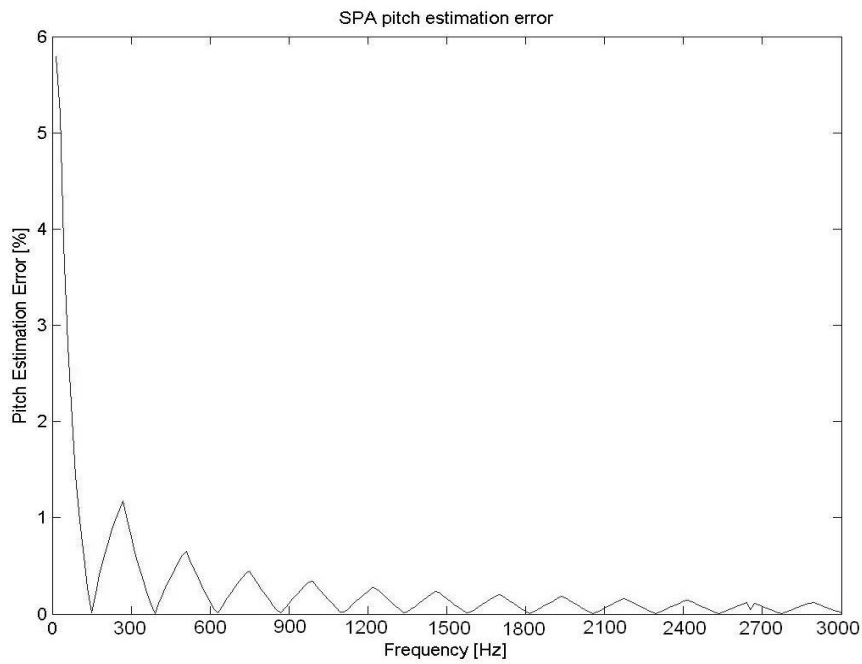Fig. 5. Pitch estimation error of the CA algorithm.

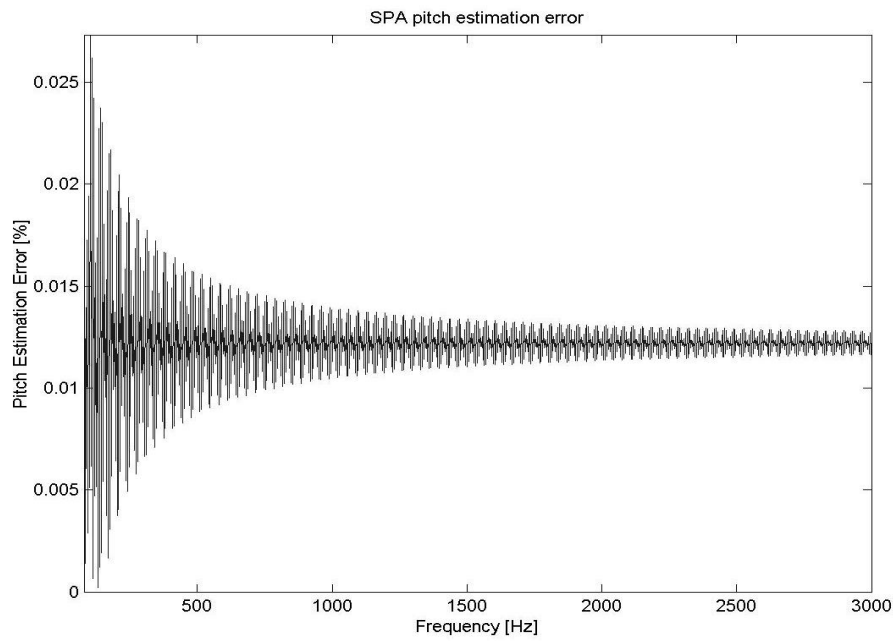Fig. 6. Pitch estimation error of the SPA algorithm (not optimized).

Fig. 7. Pitch estimation error of the SPA algorithm (2nd order polynomial interpolation).
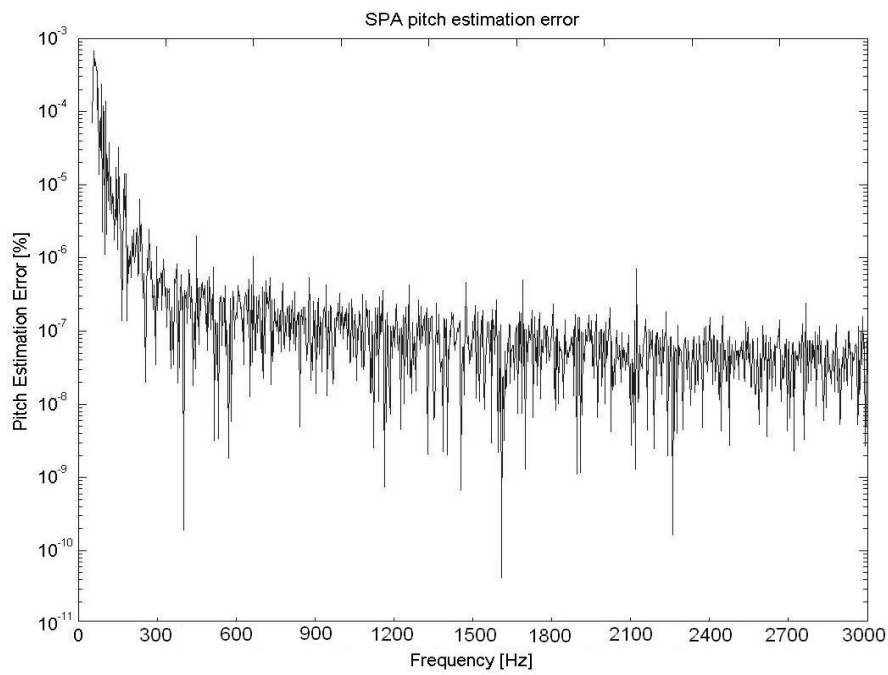

Fig. 8. Pitch estimation error of the SPA algorithm (ANN-based interpolation).

Figure 4 presents performance of the NCC algorithm, showing an increase in errors from 0.2% for the lowest frequencies to 6% for frequencies around 3000 Hz. Figure 5 presents performance of the CA algorithm. It can be observed that in this case also, error changes in frequencies are similar, however, fluctuations are more significant for frequencies over 1500 Hz. Figures 6–8 present the behavior of the SPA algorithm. Figure 6 shows estimation accuracy for the engineered algorithm without interpolating harmonic peak (i.e. frequency of the maximum value of the peak represents fundamental frequency), resulting in error equal to 5.8% for the lowest frequencies, and decreasing to 0.1% for frequencies around 3000 Hz. Figure 7 presents the improved performance of the algorithm by employing 2nd order polynomial interpolation. This results in errors of 0.027% for the lowest frequencies, decreasing to 0.007% for frequencies around 3000 Hz. Figure 8 shows performance of the ANN-based interpolation of the harmonic peak. The estimated error is equal to 0.0005% for the lowest frequencies decreasing to 0.00000013% for frequencies around 3000 Hz.

4. Time domain pitch contour correction

In some cases, transients of analyzed instrument sounds, contain only or almost only odd harmonics, therefore pitch, calculated in short terms for transient parts, can be perceived as one octave higher than pitch calculated for blocks representing steady state of the sound. The human brain seems to ignore this fact, and for a listener the perceived pitch of the whole sound is in accordance with that of the steady-state. However, blocks containing transient, duplicated in time domain, result in sound with pitch perceived as one octave higher. This observation calls for post-processing [5], i.e., time domain pitch contour correction. Optimizing pitch tracks is relatively easy, since such problems are only encountered for transient parts of musical sounds and in the majority of cases pitch contour represents the expected (perceived) fundamental frequency. In Fig. 9 one can observe that for an oboe, for one block in the transient phase, the estimated pitch is one octave higher than that estimated for the steady-state, however, the overall pitch was recognized correctly.

5. Experiments and results

In order to determine the efficiency of presented SPA, 412 musical instrument sounds were tested. Analyses of six instruments in their full scale, representing diverse groups, and one instrument with all articulation types, were carried out. Recordings of tested sounds were made in the Multimedia Systems Department of the Faculty of Electronics, Telecommunications and Informatics, of Gdańsk University of Technology, Poland [10]. Tables (Tabs. 2–4) and figures (Figs. 10–18) present estimated average pitch, note played by the instrument according to ASA standard, and the nominal frequency of the note, as specified by the ASA. Results for oboe for three types of articulations: *non legato, staccato and portato* are presented in Tables. 2–4. Results for other instruments, dynamics and articulations are presented in Figs. 10–18.

Table 2. Pitch estimation results for **oboe** (articulation: *non legato*, dynamics: *mezzo forte*).

Tone (ASA)	Estimated pitch [Hz]	Nominal freq. [Hz]	Octave error
A3#	234.24	233.08	NO
B3	245.46	246.94	NO
C4	263.22	261.63	NO
C4#	279.8	277.18	NO
D4	295.94	293.66	NO
D4#	314.52	311.13	NO
E4	332.35	329.63	NO
F4	351.04	349.23	NO
F4#	371.95	369.99	NO
G4	394.19	392	NO
G4#	417.42	415.3	NO
A4	442.4	440	NO
A4#	471.37	466.16	NO
B4	498.13	493.88	NO
C5	528.85	523.25	NO
C5#	563.3	554.37	NO
D5	597.98	587.33	NO
D5#	632.25	622.25	NO
E5	669.99	659.26	NO
F5	708.24	698.46	NO
F5#	755.94	739.99	NO
G5	799.07	783.99	NO
G5#	842.1	830.61	NO
A5	888.01	880	NO
A5#	936.42	932.33	NO
B5	997.3	987.77	NO
C6	1052.2	1046.5	NO
C6#	1124.5	1108.7	NO
D6	1185.5	1174.7	NO
D6#	1272.8	1244.5	NO
E6	1326.3	1318.5	NO
F6	1407.1	1396.9	NO
F6#	1502.1	1480	NO

Table 3. Pitch estimation results for **oboe** (articulation: *portato*, dynamics:*mezzo forte*).

Tone (ASA)	Estimated pitch [Hz]	Nominal freq. [Hz]	Octave error
A3#	234.98	233.08	NO
B3	246.48	246.94	NO
C4	263.76	261.63	NO
C4#	279.92	277.18	NO
D4	296.12	293.66	NO
D4#	313.06	311.13	NO
E4	332.96	329.63	NO
F4	352.04	349.23	NO
F4#	373.6	369.99	NO
G4	396.97	392	NO
G4#	422.38	415.3	NO
A4	447.6	440	NO
A4#	472.71	466.16	NO
B4	500.22	493.88	NO
C5	530.36	523.25	NO
C5#	564.36	554.37	NO
D5	594.88	587.33	NO
D5#	631.44	622.25	NO
E5	668.94	659.26	NO
F5	706.49	698.46	NO
F5#	753.12	739.99	NO
G5	798.01	783.99	NO
G5#	846.62	830.61	NO
A5	896.12	880	NO
A5#	947.18	932.33	NO
B5	1005.1	987.77	NO
C6	1058.4	1046.5	NO
C6#	1131.3	1108.7	NO
D6	1202.9	1174.7	NO
D6#	1296.1	1244.5	NO
E6	1435.5	1318.5	NO

Table 4. Pitch estimation results for **oboe** (articulation: double *staccato,* dynamics:*mezzo forte*).

Tone (ASA)	Estimated pitch [Hz]	Nominal freq. [Hz]	Octave error
A3#	234.39	233.08	NO
B3	245.85	246.94	NO
C4	264.03	261.63	NO
C4#	279.62	277.18	NO
D4	294.79	293.66	NO
D4#	313.57	311.13	NO
E4	331.49	329.63	NO
F4	351.44	349.23	NO
F4#	374.11	369.99	NO
G4	396.23	392	NO
G4#	422.11	415.3	NO
A4	442.7	440	NO
A4#	471.08	466.16	NO
B4	498.67	493.88	NO
C5	532.04	523.25	NO
C5#	568.96	554.37	NO
D5	598.54	587.33	NO
D5#	635.64	622.25	NO
E5	669.15	659.26	NO
F5	708.18	698.46	NO
F5#	754.25	739.99	NO
G5	798.97	783.99	NO
G5#	850.02	830.61	NO
A5	898.31	880	NO
A5#	941	932.33	NO
B5	1007.5	987.77	NO
C6	1068.4	1046.5	NO
C6#	1152.3	1108.7	NO
D6	1217.7	1174.7	NO
D6#	1298.3	1244.5	NO
E6	1425.6	1318.5	NO
F6	1494.5	1396.9	NO
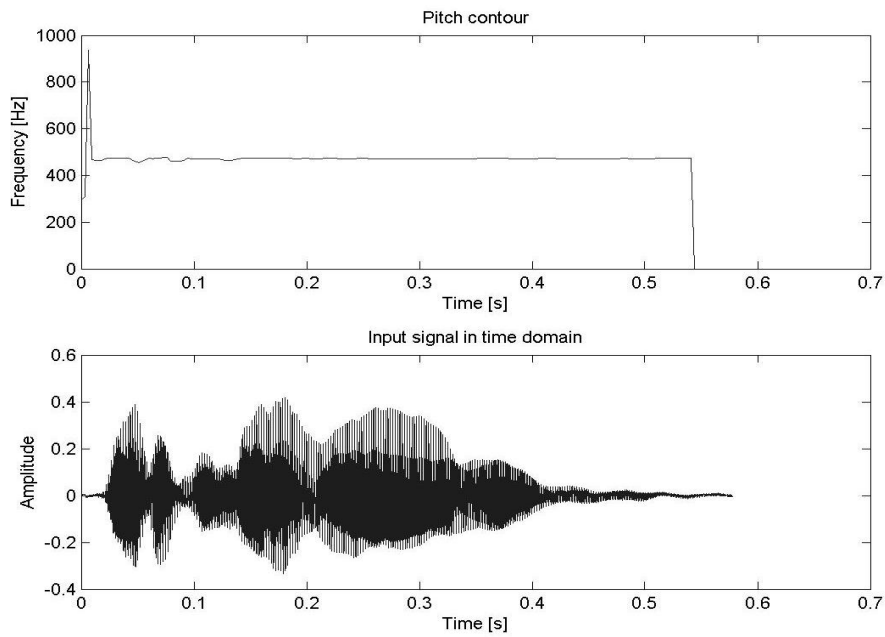
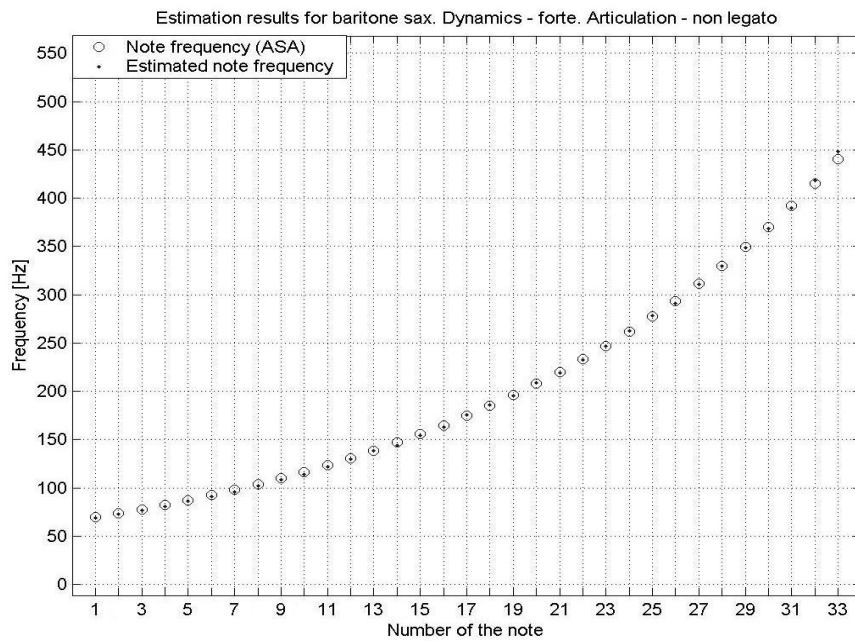Fig. 9. Octave fluctuations of pitch in transient of oboe (*non legato*).

Fig. 10. Pitch estimation results for baritone saxophone (articulation: *non legato,* dynamics: *forte*, range: C2# – A4).
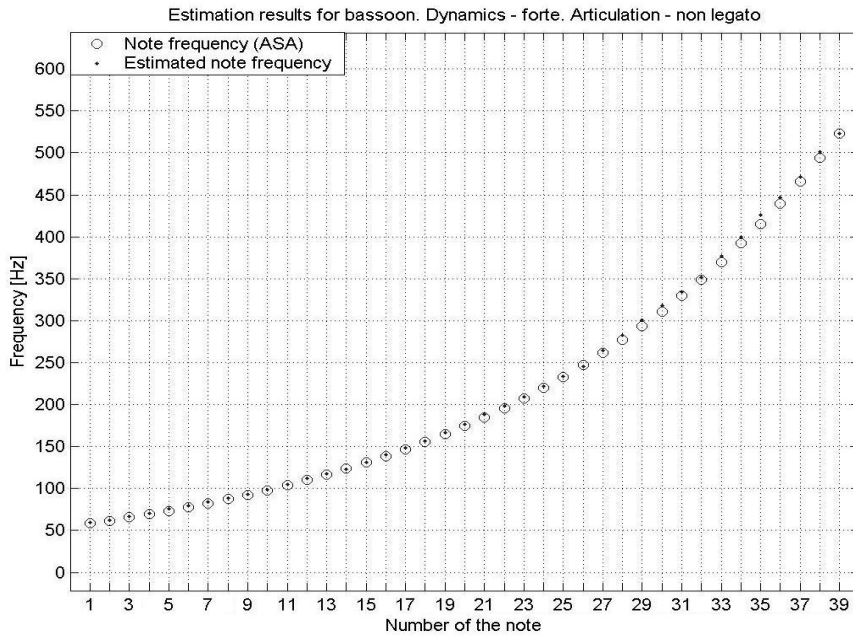
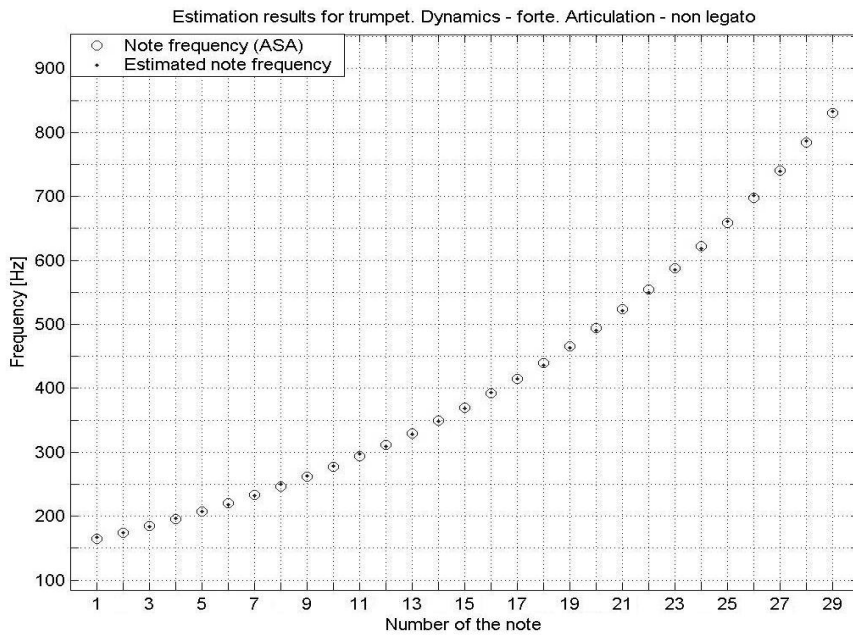Fig. 11.   Pitch estimation results for bassoon (articulation: *non legato*, dynamics: *forte*, range: A1# – C5).

Fig. 12.   Pitch estimation results for trumpet (articulation: *non legato*, dynamics: *forte*, range: E3 - G5#).
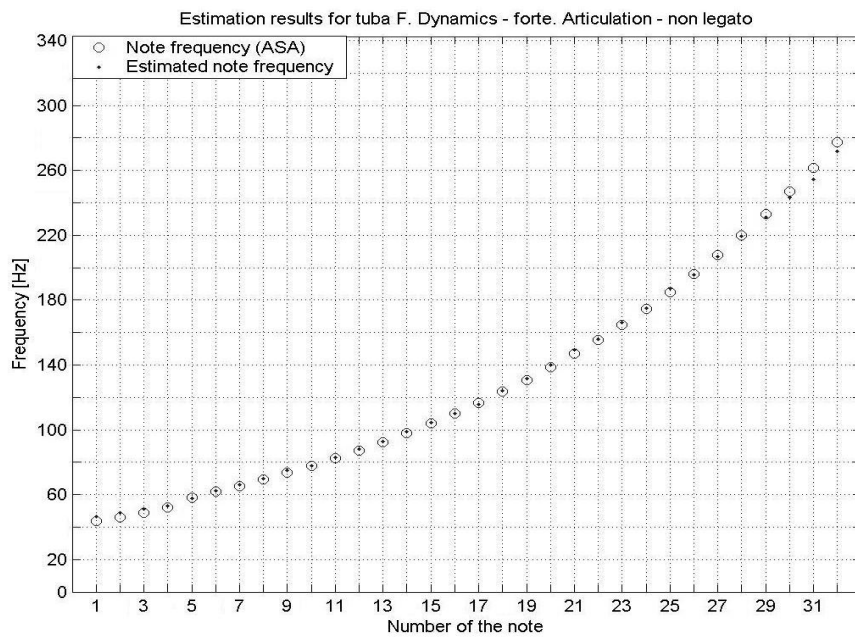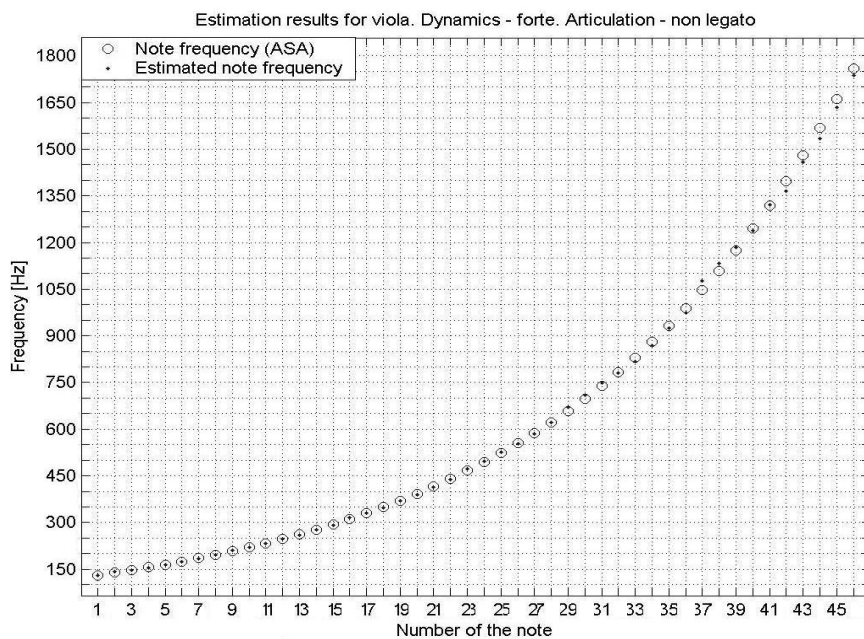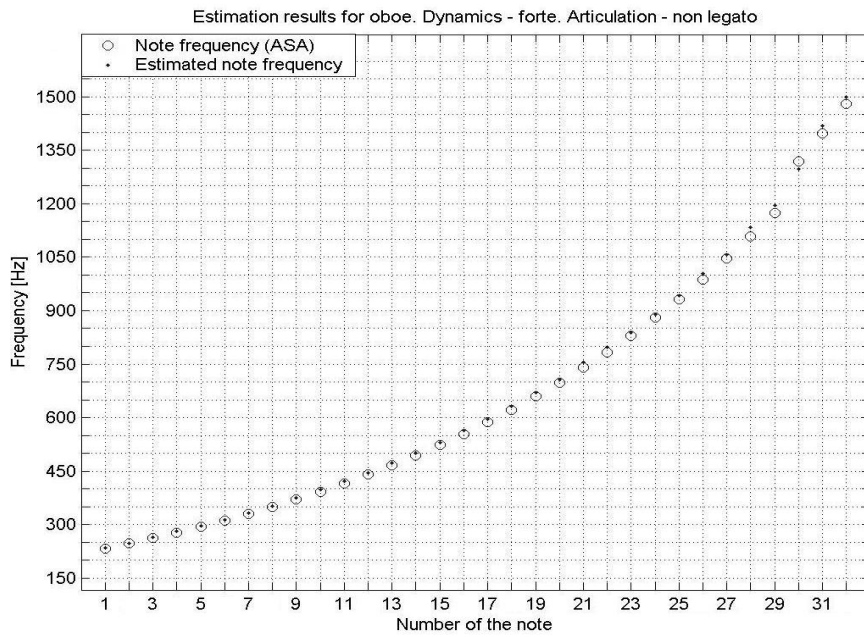
Estimation results for tuba F. Dynamics - forte. Articulation - non legato

Fig. 13. Pitch estimation results for tuba F (articulation: *non legato*, dynamics: *forte*, range: F1 - C4#).

Estimation results for viola. Dynamics - forte. Articulation - non legato

Fig. 14. Pitch estimation results for viola (articulation: *non legato*, dynamics: *forte*, range: C3 - A6).

Fig. 15. Pitch estimation results for oboe (articulation: *non legato*, dynamics: *forte*, range: A3# - F6).
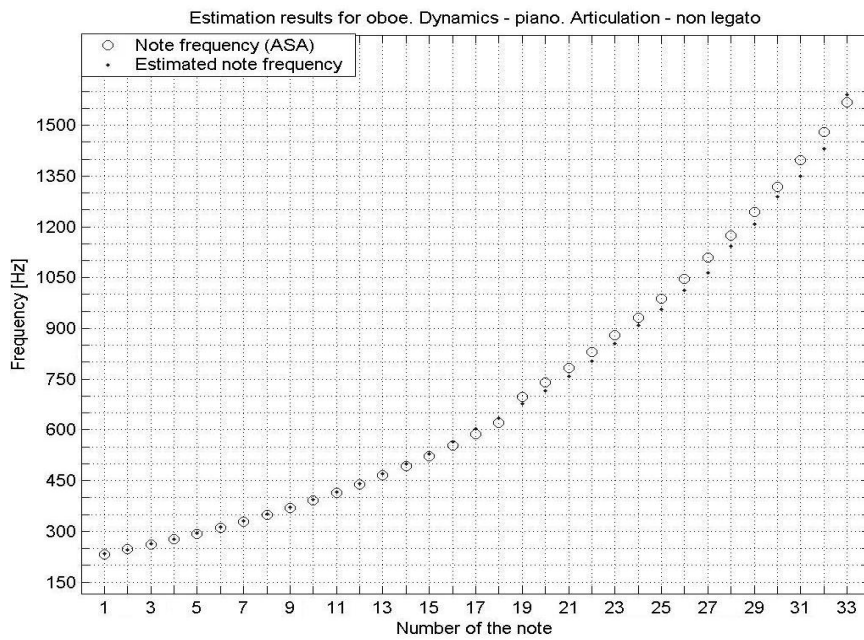

Fig. 16. Pitch estimation results for oboe (articulation: *non legato*, dynamics: *piano*, range: A3# - F6#).
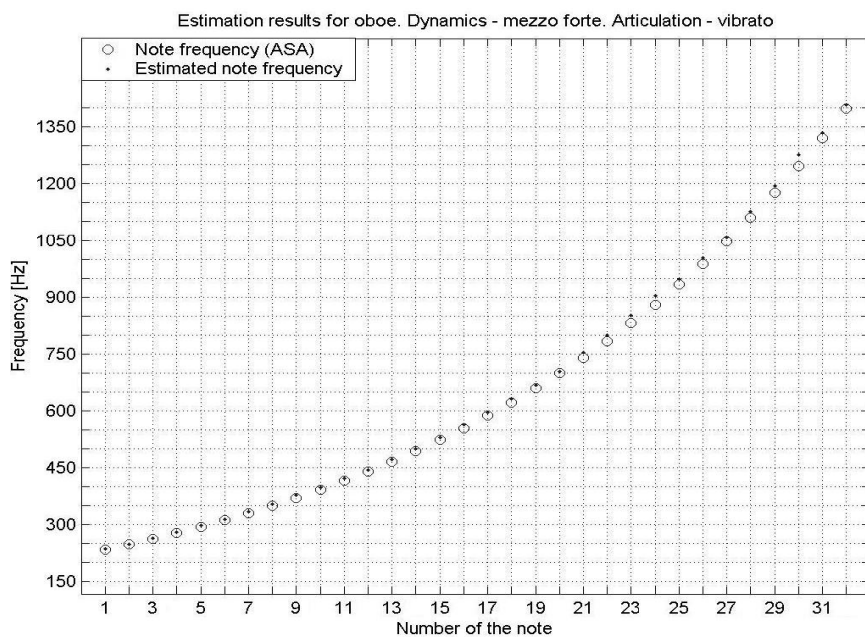
Fig. 17. Pitch estimation results for oboe (articulation: *vibrato*, dynamics: *mezzo forte*, range: A3# – F6).
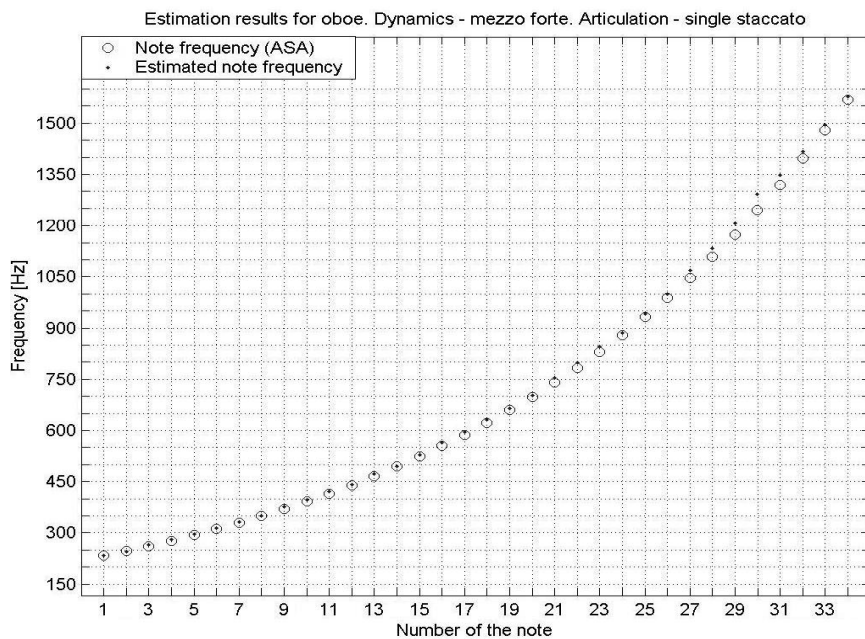

Fig. 18. Pitch estimation results for oboe (articulation: single *staccato*, dynamics: *mezzo forte*, range: A3# – G6).

As seen from tables and figures presented, no octave related errors were detected by the engineered algorithm. Different articulations and dynamics of sounds seemed not to affect the octave error estimation accuracy of the SPA. Differences, sometimes significant, between estimated pitch and tone frequency arise as the result of musicians playing solo. Moreover, instruments were not tuned to exactly the same pitch before the recordings.

6. Conclusion

The proposed algorithms have been tested on a variety of sounds with differentiated articulations and dynamics, showing high resistance to octave errors (octave error was not detected among all tested sounds). In addition, there is no limitation to harmonic sounds in the analysis (while periodicity has to be maintained), which is the case with other algorithms, such as, for example, CA and ACOLS algorithms. Moreover, energy of harmonics does not have to be concentrated around a fundamental frequency, which is an important issue for both: NCC and AMDF algorithms. The main disadvantage of the SPA presented is its limited frequency range for small window sizes (lower boundary). On the other hand, the NCC algorithm has an extended lower frequency limit. However, in case of fast pitch fluctuations of low pitched sounds, the overlap can be decreased significantly, while keeping large window sizes and resolution of calculated pitch track may be preserved.

In addition, presented algorithm accuracy optimization seems to be very effective, resulting in very precise pitch estimation. An optimized SPA algorithm gives far more precise results than classic PDAs, these characteristics may be useful in sound separation and parameterization processes.

Acknowledgment

References

[1] W. HESS, *Pitch determination of speech signal processing*, Springer-Verlag, New York 1983.

[2] A. M. NOLL, *Cepstrum pitch determination*, J. Acoust. Soc. Am., **14**, 293–309 (1967).

[3] L. R. RABINER, *On the use of autocorrelation analysis for pitch detection*, IEEE Trans. on ASSP, **25**, 24–33 (1977).

[4] X. QUIAN, R. KIMARESAN, *A variable frame pitch estimator and test results*, IEEE Int. Conf. On Acoustics, Speech, and Signal Processing, **1**, Atlanta GA, 228–231, May (1996).

[5] D. TALKIN, *A robust algorithm for pitch tracking (RAPT)*, Speech Coding And Synthesis, pp. 495-518, Elsevier, 1995.

[6] G. S. YING, L. H. JAMIESON, C. D. MICHELL, *A probabilistic approach to AMDF pitch detection*, http://purcell.ecn.purdue.edu/~speechg

[7] Y. MEDAN, E. YAIR, D. CHAZAN, *An accurate pitch detection algorithm*, 9-th Int. Conference on Pattern Recognition, Rome, Italy, **1**, 476–80, November (1988).

[8] W. ZHANG, G. XU, Y. WANG, *Pitch estimation based on circular AMDF*, ICASSP **1**, 341–344 (2002).

[9] X. MEI, J. PAN, S. SUN, *Efficient algorithms for speech pitch estimation*, Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Proc. Hong Kong, pp. 421-424, (2001).

[10] B. KOSTEK, A. CZYŻEWSKI, *Representing musical instrument sounds for their automatic classification*, J. Audio Eng. Soc., **49**, 9, 768–785 (2001).

[11] J. D. WIZE, J. R. CAPRIO, T. W. PARKS, *Maximum-likelihood pitch estimation*, IEEE Trans. of ASSP, **24**, 418–423, October (1976).

[12] J. HU, S. XU, J. CHEN, *A modified pitch detection algorithm*, IEEE Communications Letters, **5**, 2 (2001).

[13] K. KASI, S. A. ZAHORIAN, *Yet another algorithm for pitch tracking*, ICASSP, **1**, 361–364 (2002).

[14] N. KUNIEDA, T. SHIMAMURA, J. SUZUKI, *Robust method of measurement of fundamental frequency by ACOLS-autocorrelation of log spectrum*, IEEE Int. Conf. On Acoustics, Speech, and Signal Processing, **1**, Atlanta, GA, 232–235, May (1996).

[15] L. JANER, *Modulated gaussian wavelet transform based speech analyser pitch detection algorithm*, Proc. EUROSPEECH, **1**, 401–404 (1995).

[16] R. J. MCAULAY, T. F. QUATIERI, *Pitch estimation and voicing detection based on a sinusoidal speech model*, ICASSP, **1**, 249–252 (1990).

[17] L. R. RABINER, M. J. CHENG, A. E. ROSENBERG, C. A. MCGOGENAL, *A comparative performance study of several pitch detection algorithms*, IEEE Trans. on Acoustics, Speech and Signal Proc., **ASSP-24**, 5, October (1976).

[18] C. A. MCGOGENAL, L. R. RABINER, A. E. ROSENBERG, *A subjective evaluation of pitch detection methods using LPC synthesized speech*, IEEE Trans. on Acoustics, Speech and Signal Proc., **ASSP-25**, 3, June (1977).

[19] R. AHN, W. H. HOLMES, *An improved harmonic-plus-noise decomposition method and its application in pitch determination*, Proc. IEEE Workshop on Speech Coding for Telecommunications, Pocono Manor, Pennsylvania, pp. 41-42, (1997).

[20] C. D'ALESSANDRO, B. YEGNANARAYANA, V. DARSINOS, *Decomposition of speech signals into deterministic and stochastic components*, ICASSP, **1**, 760–763 (1995).

[21] S. OSOWSKI, *Artificial neural networks in algorithmic approach* [in Polish], WNT, Warsaw 1996.