On-line Empirical Mode Decomposition Biomedical Microprocessor for Hilbert Huang Transform

Nai-Fu Chang, Tung-Chien Chen, Cheng-Yi Chiang and Liang-Gee Chen
Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, Email: lgchen@cc.ee.ntu.edu.tw

***Abstract*— On-chip implementation of Hilbert-Huang transform (HHT) has great impact to analyze the non-linear and non-stationary biomedical signals on wearable or implantable sensors for the real-time applications. Empirical mode decomposition (EMD) is the key component for the HHT processor. In tradition, EMD is usually performed after the collection of a large window of signals, and the long latency may not be feasible for the real-time applications. In this work, the architecture of on-line EMD for biomedical signals is proposed. The on-line interpolation method with data reuse as well as component and iteration loop decomposition is applied to obtain low latency and low hardware cost. The first chip of EMD processor is fabricated in UMC 90nm LL process and consumes 57.3μW.**

I. INTRODUCTION

Most of biomedical signals, e.g. electroencephalogram (EEG) and electrocardiogram (ECG), exhibit significant complex behavior with strong non-linear and non-stationary properties. Hilbert-Huang transform (HHT), proposed by Huang et al. [1], is an adaptive data analysis method, composed of a decomposition method called, empirical mode decomposition (EMD), and Hilbert spectrum analysis. Recently, HHT has been widely used for biomedical engineering, such as sleep analysis [2], signal de-noising [3] and heart analysis[4], etc. Besides, hardware implementation of EMD using DSP and FPGA was proposed [5].

The VLSI on-chip system is the current trend because of the power and area constraints imposed by the wearable or implantable devices. As front-end and wireless circuits are acquirable, integrated signal processors are needed to complete the system for real-time applications. On-chip systems for some of the key signal processing algorithms, such as fast Fourier transform (FFT) and discrete wavelet transform (DWT), have been proposed [6][7]. HHT is an important signal processing tool but its on-chip system has not been implemented so far.

The key part of the algorithm is the EMD method, in which cubic spline interpolation (CSI) consumes the most computation [8]. The heavy sifting process in the EMD results in high time delays and large memory. The sequential flow and the high computation to solve the tridiagonal matrix algorithm (TDMA) for CSI cause high computation loading and high computation delays. Due to these two issues, HHT is often used off-line through personal computer. An on-line EMD with low delays and small computation resources is required for EMD microprocessor.

In this paper, the hardware architecture performing the on-line EMD for biomedical signals in real time is proposed. An on-line interpolation method with data reuse is proposed to reduce time delays, memory and computation. The EMD sifting process is decomposed to achieve low computation delays. The remainder of this paper is organized as follows. Section II briefly reviews the theory of the EMD method and the CSI algorithm. The proposed architecture is described in Section III. The implementation results are shown in Section IV. Finally, Section V describes the conclusion.

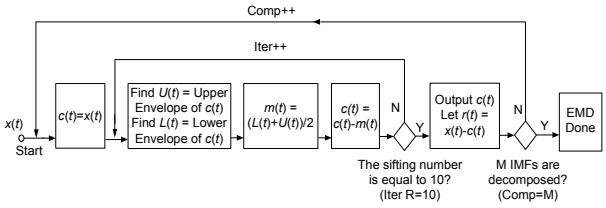


Fig. 1. The flow chart shows the EMD sifting process.

II. ALGORITHM

A. Empirical Mode Decomposition

The EMD method decomposes the complicated data set into a finite and often small number of intrinsic mode functions (IMFs). By definition, an IMF satisfies two conditions. First, the number of extrema and the number of zero-crossings must be either equal or differ at most by one in the whole data set. Another condition is that the mean value of the envelope defined by the local maxima and the envelope defined by local minima is zero at any point. Wu and Huang [9] proposed that the fixed sifting number for the decomposition is 10 to guarantee the stability and convergence of the resulting IMFs.

Fig. 1 shows the flow of EMD sifting process. Given a signal $x(t)$, the sifting process of EMD can be summarized in the following steps.

1) Identify the local maxima and minima of $x(t)$.

2) Generate the upper envelope $U(t)$ and the lower envelope $L(t)$ via CSI among all the maxima and minima, respectively.

3) Compute the average of the two envelopes. The average is defined as $m(t) = (U(t) + L(t))/2$.

4) Subtract $m(t)$ from the data to obtain an IMF candidate, that is $c(t) = x(t) - m(t)$.

5) Regard $c(t)$ as the new input and repeat the step 1-4 until 10_{th} iterations. Then, $c(t)$ can be regarded as an IMF.

6) Evaluate the residue $r(t)$ by separating the IMF $c(t)$ from original signal $x(t)$ as $r(t) = x(t) - c(t)$.

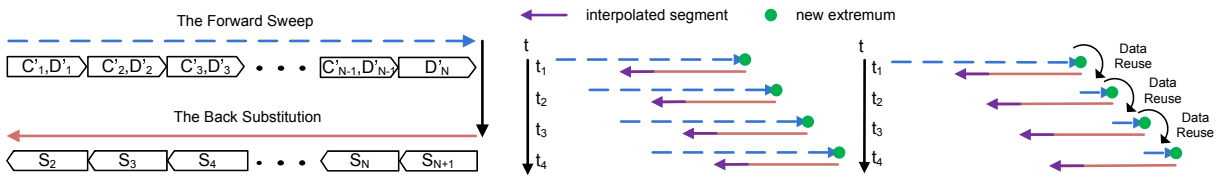
7) Take $r(t)$ as input data and repeat step 1-6, and obtain a number of IMFs until the two properties of IMF are fulfilled.

After EMD sifting process, the signal can be represented as the sums of several IMFs from the high-frequency part to the low-frequency part and the residue.

$$x(t) = r(t) + \sum_{i=1}^{M} c_i(t), \tag{1}$$

where M is the number of IMF components. In practice, M is set according to different applications and demands. For

978-1-4577-1470-2/11/$26.00 ©2011 IEEE



(a) off-line TDMA schedule with a large data window

(b) on-line TDMA schedule with a small overlapped timing-shift window

(c) the proposed schedule with data reuse

Fig. 2. The data dependencies on the forward sweep and the back substitution of the TDMA are shown in (a). For (b) and (c), the concept of on-line TDMA schedule with overlapped window and data reuse is applied on the interpolation and reduces the computation loading.

example, the number of IMFs, usually used to analyze EEG signals, is about two to five.

There are two main loops in each EMD sifting process - the component loop and the iteration loop. The data dependencies on the component loop and the iteration loop in EMD results in high time delays and large memory to store the data.

B. Cubic Spline Interpolation

CSI is applied to connect the maxima and the minima in the EMD process. Suppose a set of n maxima (minima) is denoted by m_1, m_2, ... and m_n with sample numbers t_1, t_2, ... and t_n, piecewise third degree polynomial functions are used to present the $n-1$ intervals between n maxima (minima). We can write the equations of these functions as follows,

$$M_k(t) = a_k t^3 + b_k t^2 + c_k t + d_k, \tag{2}$$

where $t \in [0, t_{k+1} - t_k]$ and $k = 1, 2, ..., n-1$. The third order polynomial functions, called cubic splines, must satisfy three constraints. m_k, $M_k^{'}$ and $M_k^{''}$ are continuous on the interval $[t_1, t_n]$.

Then, equations can be written as follow:

$$\begin{aligned} m_k &= M_k(0) = M_{k-1}(t_k - t_{k-1}) \\ M_k^{'}(0) &= M_{k-1}^{'}(t_k - t_{k-1}) \\ M_k^{''}(0) &= M_{k-1}^{''}(t_k - t_{k-1}). \end{aligned} \tag{3}$$

Assume that S_k is the function's second derivative, $M_k^{''}(0)$, and h_k is the sample number difference, $t_{k+1} - t_k$, a tridiagonal matrix can be derived,

$$\begin{bmatrix} B_1 & C_1 & 0 & \dots & 0 & 0 \\ A_2 & B_2 & C_2 & \dots & 0 & 0 \\ 0 & A_3 & B_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & B_{N-1} & C_{N-1} \\ 0 & 0 & 0 & \dots & A_N & B_N \end{bmatrix} \begin{bmatrix} S_2 \\ S_3 \\ S_4 \\ \vdots \\ S_N \\ S_{N+1} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{N-1} \\ D_N \end{bmatrix} \tag{4}$$

where

$$\begin{aligned} N &= n - 2 \\ A_k &= h_k && \text{for} \quad k = 2 \sim N \\ B_k &= 2(h_k + h_{k+1}) && \text{for} \quad k = 1 \sim N \\ C_k &= h_k && \text{for} \quad k = 1 \sim N-1 \\ D_k &= 6\left(\frac{m_{k+2} - m_{k+1}}{h_{k+1}} - \frac{m_{k+1} - m_k}{h_k}\right) && \text{for} \quad k = 1 \sim N. \end{aligned} \tag{5}$$

The values of S_k can be solved by the TDMA with two steps. The first step consists of modifying the coefficients as follows, denoting the new modified coefficients with primes

$$C_k^{'} = \begin{cases} \frac{C_k}{B_k} & \text{for } k = 1 \\ \frac{C_k}{B_k - C_{k-1}^{'} A_k} & \text{for } k = 2 \sim N-1 \end{cases} \tag{6}$$

and

$$D_k^{'} = \begin{cases} \frac{D_k}{B_k} & \text{for } k = 1 \\ \frac{D_k - D_{k-1}^{'} A_k}{B_k - C_{k-1}^{'} A_k} & \text{for } k = 2 \sim N \end{cases} \tag{7}$$

This is the forward sweep. The solution is then obtained by the back substitution:

$$S_k = \begin{cases} D_{k-1}^{'} & \text{for } k = N+1 \\ D_{k-1}^{'} - C_{k-1}^{'} S_{k+1} & \text{for } k = N \sim 2. \end{cases} \tag{8}$$

Then, the coefficients of piecewise polynomial functions can be derived by the following equations.

$$\begin{aligned} a_k &= \frac{S_{k+1} - S_k}{6h_k} \\ b_k &= \frac{S_k}{2} \\ c_k &= \frac{m_{k+1} - m_k}{h_k} - \frac{2h_k S_k + h_k S_{k+1}}{6} \\ d_k &= m_k \end{aligned} \tag{9}$$

To solve the coefficients of the splines, two additional spline boundary conditions are required. The second derivatives of the two boundary extrema, S_1 and S_n are set to zeros. Finally, the interpolated samples in an interval can be computed by substituting the corresponding t into eq. 2.

Fig. 2a shows the data dependencies on the forward sweep and the back substitution in the TDMA for CSI, which cause high computation loading and high computation delays.

III. VLSI Architecture Design

A. Proposed On-line Interpolation Method with Data Reuse

Low power consumption, miniaturized area and low latency are three primary issues for EMD biomedical microprocessor. In general, off-line EMD processing is executed on the signals with a large window length. It ensures that there are enough extrema to compute CSI and produce meaningful IMFs. However, to store a large amount of biomedical data and perform EMD afterwards will require a large memory and consume a lot of power and area. Moreover, the time delay to collect the biomedical signals is not allowed under the on-the-fly processing requirement. On-line EMD processing, the processing along with the input of signals, is a must on HHT hardware for real-time applications.

Fig. 2a shows the off-line TDMA schedule with a large data window. Because of the assumptions on boundary conditions of CSI, the boundary intervals of the envelope might have errors. Due to the data dependencies of EMD sifting process, these errors would propagate between IMFs and affects accuracy of the IMF results. The results could be completed through overlapping the windows within which the interpolated values are reliable. Fig. 2b shows the on-line TDMA schedule with a small overlapped timing-shift window. Because the middle spline is influenced less by the boundary errors, only the middle spline for a number of extrema CSI is adopted for interpolating the upper envelope $U(t)$ and the lower envelope $L(t)$. There is a tradeoff between latency and accuracy regarding the number of extrema CSI. Larger number of CSI also contributes to larger computation loading. For EEG signals, the number of extrema CSI is 8 to strike low latency and accuracy [10]. However, high computation loading is required because the forward sweep and the back substitution are performed several times for the spline coefficients. Fig. 2c shows the concept of data

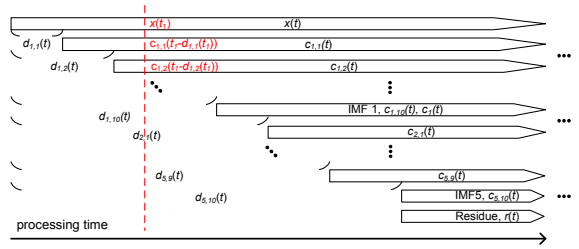


Fig. 3. The schedule of the EMD processor with the number of IMF components $M = 5$ and the sifting number $R = 10$.

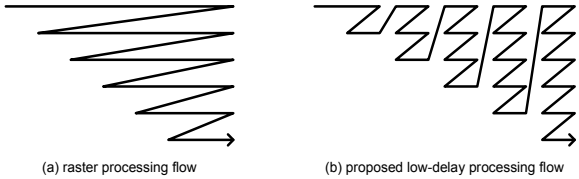


Fig. 4. In the proposed low-latency processing flow, the computation of the IMF candidates and the IMF components is repeatedly performed every time when a new original sample is input.

reuse. The previous forward sweep coefficients are stored. Then, only the forward sweep coefficients of last two extrema are calculated and combined with the previous coefficients. The computational loading could be decreased significantly by reusing the coefficients in the forward sweep. Otherwise, the effect of boundary propagation of one side is decreased as more forward data is collected.

B. Analysis of Data Dependencies for Component and Iteration Loop Decomposition

Fig. 3 shows the schedule of EMD process with the number of IMF components $M = 5$ and the sifting number $R = 10$. The 1^{st} candidate of the first IMF component, $c_{1,1}(t)$, is calculated with delay $d_{1,1}(t)$ due to the interpolation method to interpolate the middle spline between detected extrema. When $c_{1,1}(t)$ is computed, the 2^{nd} candidate of the first IMF component, $c_{1,2}(t)$, is generated with delay $d_{1,2}(t)$ larger than $d_{1,1}(t)$. While the first IMF component $c_1(t)$, is generated, the first IMF component subtracted from $x(t)$ is regarded as the input signal and used to produce the candidates of the second IMF component.

As shown in Fig. 4a, the raster processing flow to compute the IMF candidates and the IMF components with a window length one by one results in high time delays. In order to lower time delay and achieve the on-the-fly processing, the component loop and the iteration loop are decomposed. Fig. 4b shows the proposed low-delay processing flow. The original biomedical signal is pushed into the hardware sample by sample, and the computation of the IMF candidates and the IMF components is repeatedly performed every time when a new sample is input.

C. Final Architecture

Fig. 5 shows the architecture performing the on-line EMD process. i and j in the red words are the component index and the iteration index, respectively. This architecture can support the proposed on-line interpolation method with data reuse as well as the component and iteration loop decomposition.

The three buffers and the PUs in Fig. 5 are designed to support the on-line interpolation method with data reuse. One buffer is used to store the IMF candidates $c_{i,j}$. Another buffer

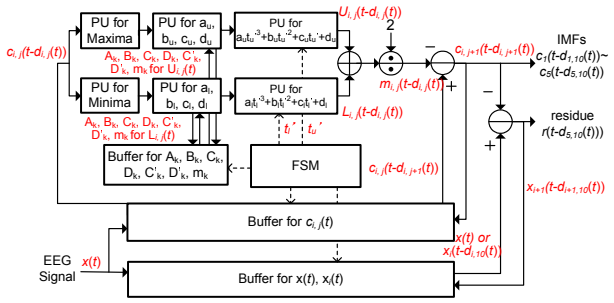


Fig. 5. The proposed architecture performing the on-line empirical mode decomposition. i and j in the red words are the component index and the iteration index, respectively. k represents the coefficient index.

is used to store the the input signals $x(t)$ and $x_i(t)$. The other buffer is used to store the previous spline coefficients for data reuse. Two sets of processing units (PUs) are used to compute the upper and the lower envelopes respectively. In each set of PUs, the first-stage PU detects whether there is a new extremum or not. If a new extremum is detected, the second-stage PU derives the available spline coefficients a_k, b_k, c_k, and d_k according to eq. 9 through computing new intermediate coefficients A_k, B_k, C_k, D_k, $C_k^{'}$ and $D_k^{'}$ as well as using previous intermediate coefficients from buffer. Afterwards, the last-stage PUs in the two sets generate the interpolated samples of available upper and lower envelopes by substituting the corresponding values of t into eq. 2. Then, an addition and a division are applied to compute the average. The average subtracted from the previous IMF candidate is the next IMF candidate and output if it's an IMF component.

The FSM controls the PUs and the buffers with the component and iteration loop decomposition. As a new original sample is input, the computation of each IMF candidate and each IMF component is distributed for a certain number of cycles. In each iteration, the FSM generates the corresponding $t_u^{'}$ and $t_l^{'}$ values for the upper envelope and the lower envelope according to the interpolating samples. The FSM also gets and stores the corresponding coefficients, IMF candidates and input signals from and to the buffers. When there's no extremum detected in the iteration, the FSM waits for the next iteration, the PUs are turned off by a gated clock.

IV. IMPLEMENTATION RESULTS

A. Precision Discussion

Precision issue on fixed-point calculation is important for the hardware implementation. Unlike the processing in the software system, it may not be efficient to do a floating-point calculation. For the fixed-point hardware, there's a tradeoff between accuracy and hardware resources. With a large bit width, the requirement on large memory for buffers and large area for PUs results in high power consumption and thus violates our design spirit. The large parts of the proposed EMD processor are the buffers for the previous IMF candidates and input signals. Fig. 6 shows the analysis on the precision error of the IMF components using the EEG data with 256 sps sampling rate. The precision error decreases as the increase of the bit width. The error of later IMFs is larger due to the error propagation between IMFs. The precision of the hardware can be decided according to different biomedical signals and applications.

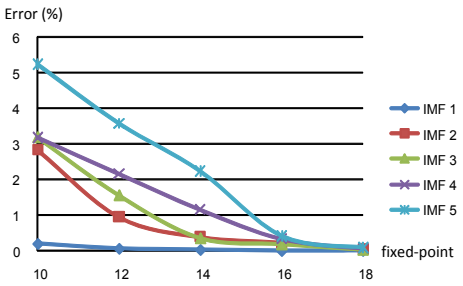


Fig. 6. The precision error of the IMF components is analyzed using the EEG data with 256 sps sampling rate.

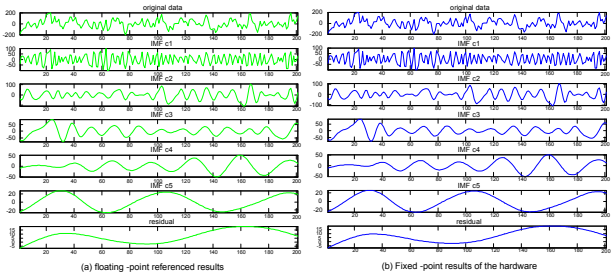


Fig. 7. The results of five IMFs and residue after EMD on the EEG signal.

B. Implementation Results

We use the UMC 90 nm 1P9M LL CMOS process to implement the on-line biomedical EMD processor. The processor decomposes the EEG signal with 256 sps sampling rate into five IMFs and residue, while the number of IMFs and the sifting number are 5 and 10, respectively. Fig. 7 shows the IMF results after EMD on the EEG signal. The time delays of the first three IMF components are 0.7, 2.0 and 4.7 seconds, respectively. Fig. 8 shows the chip photo. The implementation result is summarized (see Table I). The chip area is 2.11 mm^2 and the memory size is 24.22kB with 16 bit fixed-point precision. The system operation frequency is 522.24 kHz with 256 Hz EEG sampling rate. The total power is 57.3 μW, and the supply voltage of the chip is 1.0 Volt. While the system operation frequency is at low clock rate, the majority of power consumption comes from the leakage power, 75.37% of the total power. The number of decomposed IMFs M, the sifting number R and the fixed-point precision could be set depending on the application requirement. When M and R increase to m times and r times, the buffer size will increase to $m \times r$ times. The number of PUs will increase to p times with $\frac{m \times r}{p}$ times of the operation frequency.

C. Parallel Extension

The implemented on-line EMD processor is a prototype for a specific specification for HHT. The proposed architecture can be reused to support other specifications. Recently, multi-channel biomedical signal processing may also be required for some biomedical applications. For supporting multi-channel computing, one way is to stack the architectures and each architecture executes one channel EMD processing. Parallel architectures also could be applied to the noise-assisted EMD algorithm - ensemble empirical mode decomposition (EEMD) [11]. The original signal is added by different independent white noises to a set of ensemble signals and the EMD is applied to each ensemble signal. If n ensemble signals are used, n architectures could be applied and each architecture executes the EMD process of one ensemble signal. Besides the multi-channel processing and the EEMD, the architecture could also support paralleling of the component loop and the iteration loop. Several sets of PUs of the architecture could be stacked and each set of PUs processes certain IMFs candidates and IMF components through careful scheduling.

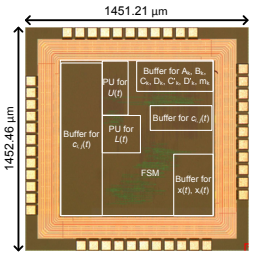


Fig. 8. The chip photo of proposed on-line EMD biomedical processor in 90nm Low Leakage CMOS process.

TABLE I
IMPLEMENTATION RESULTS OF ON-LINE EMD BIOMEDICAL PROCESSOR IN 90NM LL CMOS PROCESS

Process	$90nm$ 1P9M Low Leakage CMOS
Supply Voltage	1.0 Volt
Core Area (Memory %)	1.02 mm^2 (37.1%)
Chip Area	2.11 (1.45x1.45) mm^2
Operation Frequency	522.24 kHz
Power Consumption	57.3 μW
Computation Capability	Realtime EMD processing for 1 channel EEG (@256Hz) into five IMFs and residue

V. CONCLUSION

In this paper, the architecture design of on-line biomedical EMD processor is presented for the real-time IMF computations. The on-line interpolation method with data reuse and the component and iteration loop decomposition is designed to save memory, decrease computation loading and achieve low latency. The first EMD microprocessor is implemented in 90 nm LL process with 1.02 mm^2 core area and 57.3μW power consumption.

REFERENCES

[1] N. E. Huang *et al.*, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Royal Society*, vol. A, no. 454, pp. 903–995, 1998.

[2] L. Causa *et al.*, "Automated sleep-spindle detection in healthy children polysomnograms," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 9, pp. 2135–2146, 2010.

[3] B.-V. Manuel *et al.*, "Ecg signal denoising and baseline wander correction based on the empirical mode decomposition," *Comput. Biol. Med.*, vol. 38, pp. 1–13, 2008.

[4] J. C. Echeverria *et al.*, "Application of empirical mode decomposition to heart rate variability analysis," *Med. Biol. Eng. Comput.*, vol. 39, no. 4, pp. 471–479, 2001.

[5] M.-H. Lee *et al.*, "Hardware implementation of emd using dsp and fpga for online signal processing," *IEEE Trans. Industrial Electronic*, vol. 58, no. 6, pp. 2473–2481, 2011.

[6] T.-C. Chen *et al.*, "1.4μw/channel 16-channel eeg/ecog processor for smart brain sensor soc," in *Symp. on VLSI Circuits*, Hawaii, June 2010, pp. 21–22.

[7] A. M. Kamboh *et al.*, "Area-power efficient vlsi implementation of multichannel dwt for data compression in implantable neuroprosthetics biomedical circuits and systems," *IEEE Trans. on Biomed. Circuits and Systems*, vol. 1, no. 2, pp. 128–135, 2007.

[8] L. Wang *et al.*, "Hardware-accelerated implementation of emd," in *IEEE Biomed. Eng. Inf.*, Yantai, Oct. 2010, pp. 912–915.

[9] Z. Wu and N. E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," in *Proc. Roy. Soc. Lond.*, A460, Dec. 2004, pp. 1597–1611.

[10] N.-F. Chang *et al.*, "Cubic spline interpolation with overlapped window and data reuse for on-line hilbert huang transform biomedical microprocessor," in *IEEE EMBC*, Boston, 2011, pp. 7091–7094.

[11] Z. Wu and N. Huang, "Ensemble empirical mode decomposition: a noise assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 1, pp. 1–41, 2009.