Keywords: query language, data stream processing database management system, fetal monitoring.

Janusz JEŻEWSKI*, Michał WIDERA*, Ryszard WINIARCZYK#,
Adam MATONIA*, Tomasz KUPKA*, Adam GACEK*

DATA STREAM PROCESSING IN FETAL MONITORING SYSTEM: II. APPLICATION TO SIGNAL PROCESSING

Centralised fetal monitoring system belongs to signal processing system class. The main functions of the system are signal acquisition from bedside monitors, on-line trace analysis and dynamic presentation of incoming data. Collected data set is controlled by centralised application. Relational database management system can't process sampled at high frequency biomedical signals on-line. Therefore, we decide to build our own data management system dedicated to stream processing that support continuous queries. This paper describes a method of building a query plan based on proposed algebra. The presented example of application enables implementation of algorithm determining long and short term indices for fetal heart rate variability assessment on the basis of declarative query language. Our solution enables to define query based on data streams that makes the updated answers currently available.

1. INTRODUCTION

Centralised fetal monitoring system belongs to signal processing system class. We have worked on development of such system for many years that has resulted of building commercially available MONAKO system [1,2]. The main functions of the system are signal acquisition from bedside monitors, on-line trace analysis and dynamic presentation of incoming data. All biomedical signals are controlled by centralised application and stored in append-only disk files. Transmitted signals are combined into uniform data stream. System alert clinical personnel when any abnormal signal pattern is detected.

Classical signal processing assumes that each part of signal can be presented as time windows in buffer. Processing is realized by procedural high-level languages. Source and computed data are stored finally in static objects (i.e. disk files). The main disadvantage of this approach is extremely complicated signal processing algorithms. Minor disadvantages are: available resources are not controlled, system architecture is not scaled easily and recorded data are not useable for second party applications. These problems can be solved by application of data management system.

The selection of an appropriate database management system is determined by the specific character of recorded data. There are not commercial database management systems well suited for signal processing. Relational database management system can't process sampled at high frequency

* Institute of Medical Technology and Equipment, 118 Roosevelt St, 41-800 Zabrze, POLAND
tel./fax: (32) 2716013/2712312, e-mail: michal@itam.zabrze.pl
Institute of Theoretical and Applied Informatics, 5 Bałtycka St, 44-100 Gliwice, POLAND
tel./fax: (32) 2317319/2317026, e-mail: office@iitis.gliwice.pl

biomedical signals on-line. Additionally relational systems have limited possibilities of fast recording data in big quantities. Therefore, we decide to build our own data management system dedicated to stream processing that support continuous queries [3]. Our goal is to develop flexible and more efficient solution – declarative query language for signal processing. Recorded data should be controlled by database management system, easy available for other systems. Notation form of signal processing algorithms will be simple. Available system resources will be managed by database management system.

The base of our research is stream processing issues [4]. The data stream is an unbounded bag of elements (a,t) where the first element contains measured value and de second – time of occurrence. Most of all collected data by MONAKO are in form of time series – e.g. uniform data stream. Developed algebra and query language will help us simple record of signal processing algorithms. Presented solutions work quite proper but are not able to perform signal processing required in present-day biomedical applications [5,6]. Declarative query language for signal processing needs specific operators. For instance, relational full join operator is useless considering its realisation time. Therefore our conception is near to reduced instructions set in query language.

Created queries must return continuously updated answers. This kind of query is called continuous query. We decide to build our own data management system dedicated to signal processing that support continuous query. Ongoing research on data stream management system has not provided any sufficient and universal solutions so far. The main stream and signal computation method is sliding window technique. It is a commonly used way of presenting selected part of the data stream and signal in time [7,8]. This paper describes a method of building a query plan based on proposed algebra. The presented example of application enables implementation of algorithm determining long and short term indices for fetal heart rate variability assessment on the basis of declarative query language.

2. DATA STREAM AND CONTIONOUS QUERY LANGUAGE

A stream can be considered as a set (multiset to be precise) of pairs (s,t), indicating that a tuple s arrives on the stream at time t i.e. the first element contains measured value and the second – time of occurrence. In the MONAKO system the biomedical data streams can be presented in the form of time series. Time series are in the form of a bag of elements $(\{a_n\}, \Delta)$. Where the first element is data sequence and the second is a real number that determines time interval between the consecutive elements of the sequence. Every time series can be described with the help of data stream, however both definitions are not equivalent to each other. We found additionally for now, that inout order is assumed and hold by recording system. Streams have the notion of an input order, they are unbounded, and they are append-only. By data stream schema A we will understand the list of attributes of individual elements (tuples) of sequence $\{a_n\}$. The schema is written in the following way $A(A_1,A_2,A_3)$. Where A_1,A_2,A_3 represent areas where data are stored. It has to be noted that the relation of order within the list is compulsory. The order presented in the schema is binding. Considering data stream A as a bag of elements $(\{a_n\},\Delta_a)$ and assuming that all the operations realized in database management system will refer to set of data streams we determined the following set of operations necessary to implement the analytical procedures in MONAKO system: Interlace, Deinterlace, Sum, Difference, Aggregation/Serialisation, Projection and Selection. (Table I)

Table I. Algebraic operations

Operation name	Algebraic notation
Interlace	C := A # B
Deinterlace	A := C & Δb;B
Sum	C := A + B
Difference	A := C – (Δ_a,Δ_b)
Aggregation/Serialisation	B := AGSE(A,(A1,A2,..),step)
Projection	A' := A(A1,A2,...,An) => (A1,A2,...,Am)
Selection	C := A(A1,A2,...,An), XΘY

Presented data set and operators state the base of Algebra. Some of them are analogous to operators presented in Aurora query algebra [6]. The current draft design of our query language is presented in Table II. Syntactically, our query language is based on SQL but disallowing the WHERE clause. All presented operations including window specification applied by AGSE operation are described in previous part of this article.

Table II. Operations and example notations in continuous Query Language

Operation	Example in continuous query language
Interlace	`SELECT a,b AS C FROM A#B`
Deinterlace	`SELECT a AS A FROM C&2`
Sum	`SELECT a,b AS C FROM A+B`
Difference	`SELECT a AS A FROM C-(1,2)`
Aggregation/Serialisation	`SELECT AGSE(C,NUMBER<10>,1) FROM C`
Projection	`SELECT a FROM C`
Selection	`SELECT a FROM C FILTER C BY a>10`

As example we next present construction of query plan based on presented assumptions. In Relational DBMSs, all operators are pull-based: an operator requests data from the plan only when needed. In contrast, stream operators consume data pushed to the system by the external devices. Operators should by schedule to minimize queue size and queuing delays. Another problem is continuous query plan based on actual and historical data.

3. APPLICATION

One of the most important features of physiological FHR trace is that the intervals between fetal heart beats permanently undergo the small changes. Changes of the duration of successive cardiac cycles are determined as short-term variability (STV). The changes of STV direction and value, causing an FHR oscillation in relation to its mean value, are denoted as long-term variability (LTV). In a real FHR trace, both types of variability coexist showing the mutual relationship. In order to assess these values, the variability indices have been introduced. Procedure of the indices computation is performed in buffers. The computation of each index occurs every minute and requires the collection of 240 FHR signal samples. Since the indices are defined on a basis of signal averaged over 2.5 s periods, their presampling is necessary. To this end, the samples are grouped in tens, and for each group the mean value is determined thus creating the Fhr_Avg table which is a common basis for further determination of LTV and STV indices. When determining the STV, Fhr_DivAvgMonitor table is created that contains the differences between successive elements of Fhr_Avg table. STV value, placed in the window with results of analysis, presents a mean value from all elements of the Fhr_DivAvgMonitor table. To compute LTV there are determined the minimal and maximal elements within Fhr_Avg table, whereas the resulting LTV demonstrates the difference between these values. Consecutive LTV and STV values create a sequence of currently added values. These values are being presented together with FHR signal in a form of histograms including the averaged information on changes in a fetal heart rhythm. The diagrams are updated every minute.

MONAKO system records input stream (InStream) from 8 bedside fetal monitors. Elements of this stream are added every 0.125 seconds. Every tuple contains information about monitor number and values of measured signals: fetal heart rate (FHR), oxygen saturation (OXY), and uterine contraction (UC). Our goal is to present the sequence of necessary operations (query plan) with the help of previously defined operations, enabling calculation so-called long- and short-term FHR variability indices (LTV and STV) presented in MONAKO system for the monitor number 1. Data stream interval Δ in MONAKO system in nominated by a unit of time calculated in seconds. Input stream schema InStream,0.125s is presented as follows:

$$\text{InStream}\ (\ \text{ID\_FMonitor}, \text{FHR}_1, \text{FHR}_2, \text{FHR}_3, \text{FHR}_4, \text{OXY}, \text{UC}\ ) \tag{1}$$

First operation is selection of tuples belonging to fetal monitor number 1. This operation is recorded in the following way:

$$\text{FMonitor1} := \text{InStream}\ (\text{ID\_FMonitor}, \text{FHR}_1, \text{FHR}_2, \text{FHR}_3, \text{FHR}_4, \text{OXY}, \text{UC}),\ \text{ID\_FMonitor} = 1 \tag{2}$$

Resulting stream contains tuples coming exclusively from monitor number 1. The interval Δ of resulting stream is 1s. Deinterlace operation of the given streams for this data stream cannot be applied. We have to take into consideration input stream including data from only few fetal monitors that can be connected.
In this way we receive data stream FMonitor1,1s whose consecutive tuples are added once in every second. Next operation is projection operation:

$$\begin{aligned} \text{FHR4\_ FMonitor1} := {} & \text{FMonitor1}\ (\text{ID\_FMonitor, FHR1, FHR2, FHR3, FHR4, OXY}) \\ & \Rightarrow (\text{FHR1, FHR2, FHR3, FHR4}) \end{aligned} \tag{3}$$

Interval Δ remains unchanged and equals 1s. Next step is serialization operation of stream FHR4_FMonitor1,1s.

$$\text{FHR\_FMonitor1} := \text{FHR4\_ FMonitor1}\ (\ \text{DOM(FHR)},\ 1\ ) \tag{4}$$

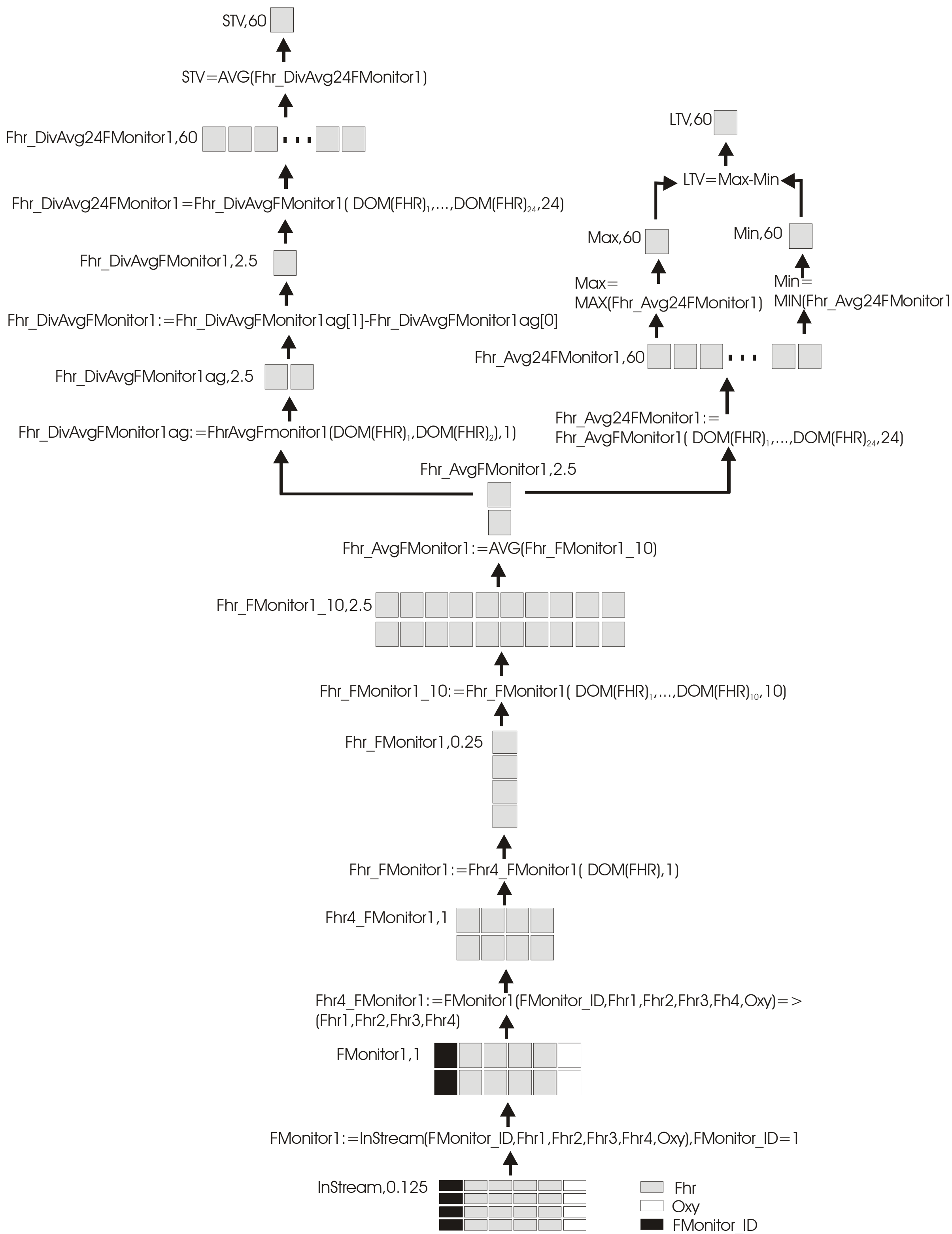

Fig. 1 Query plan for STV and LTV index computation.

After this operation has been finished we receive data stream FHR_FMonitor1,0.25s. Notation DOM(FHR) means domain of FHR, i.e. in our example DOM(FHR) means NUMBER. Its elements contain atomic, measured values of fetal heart rate (FHR). 240 samples of consecutive values of fetal heart rate are required to determine long- and short-term variability indices.

Because variability indices are defined on the basis of average signal for the period of 2.5 seconds, it is necessary to sample it. Considering this, samples are grouped in 10s. An average value is determined from every appointed group. The created table of 24 values is the direct base to determine the searched indices. The described process of determining indices should be started from data streams aggregation FHR_FMonitor1,0.25s.

$$\text{FHR\_FMonitor1\_10} := \text{FHR\_FMonitor1}\ (\ (\ \text{DOM(FHR)}_1,\ldots,\text{DOM(FHR)}_{10}\ ), 10\ ). \tag{5}$$

$\text{DOM(FHR)}_1,\ldots,\text{DOM(FHR)}_{10}$ record means data schema consisted of 10 fields including consecutive values of FHR. Every stream tuple FHR_FMonitor1_10,2.5s includes data from 10 recorded by the monitor measurements of fetal heart rate values e.g. (FHR(n+1),FHR(n+2),..., FHR(n+10)). As a result of algebraic simplification it is feasible to link two previous operations in one.

$$\text{FHR\_FMonitor1\_10} := \text{FHR4\_ FMonitor1}\ (\ (\ \text{DOM(FHR)}_1,\ldots,\text{DOM(FHR)}_{10}\ ), 4\ ) \tag{6}$$

For further calculation it is necessary to create stream including average from determined tuples – with this end in view we carry out the following operations:

$$\text{FHR\_AvgFMonitor1} := \text{AVG}\ (\ \text{FHR\_FMonitor1\_10}\ ) \tag{7}$$

Stream whose schema is as follows is created: FHR_AvgFMonitor1 (DOM(FHR)), its elements are added once in every 2,5 seconds. Every following stream tuple FHR_AvgFMonitor1,2.5s includes information about average from 10 consecutive measured FHR values. Short term variability index (STV) is determined on the basis of 24 consecutive elements of the stream, whose tuples store the difference of consecutive elements of the stream FHR_AvgFMonitor1,2.5s. The following operations enable to create the stream whose consecutive elements present differences of consecutive elements of the stream FHR_AvgFMonitor1,2.5s.

$$\begin{aligned} \text{FHR\_DivAvgFMonitor1ag} &:= \text{FHR\_AvgFMonitor1}\ (\ (\text{DOM(FHR)}_1,\text{DOM(FHR)}_2\ ),\ 1\ ) \\ \text{FHR\_DivAvgFMonitor1} &:= \text{FHR\_DivAvgFMonitor1ag.FHR2} \\ &\quad - \text{FHR\_DivAvgFMonitor1ag.FHR1} \end{aligned} \tag{8}$$

Stream schema FHR_DivAvgFMonitor1,2.5s is identical with schema FHR_AvgFMonitor1,2.5s. Determination of short term variability index requires calculation of the average from consecutive 24 elements of stream FHR_DivAvgFMonitor1,2.5s. The next step is the calculation of the average from every received tuple. First operation can be realized by stream aggregation FHR_DivAvgFMonitor1,2.5s.

$$\text{FHR\_DivAvg24FMonitor1} := \text{FHR\_DivAvgFMonitor1}\ (\ \text{DOM(FHR)1},\ldots,\text{DOM(FHR)24}\ ,\ 24\ ) \tag{9}$$

As a result of aggregation the stream is created FHR_DivAvg24FMonitor1,60s including 24 elements in every tuple. The function determining the average occurs in the second operation.

$$\text{STV\_FMonitor1} := \text{AVG(}\ \text{FHR\_DivAvg24FMonitor1}\ \text{)} \tag{10}$$

After these operation have been finished stream STV_FMonitor1s,60s includes consecutive determined short term variability index. Consecutive elements to this stream are added once in every minute – so the interval Δ equals 60s.
Long term variability index LTV is determined on the basis of stream FHR_Avg24FMonitor1,2.5s. In order to determine this index it is necessary to create data stream whose elements will contain the difference between minimal and maximal value of stream elements FHR_Avg24FMonitor1,2.5s. Formal record of this operation looks as follows:

$$\text{LTV\_FMonitor1} := \text{MAX(}\ \text{FHR\_Avg24FMonitor1}\ \text{)} - \text{MIN(}\ \text{FHR\_Avg24FMonitor1}\ \text{)} \tag{11}$$

Streams LTV_FMonitor1,60s and STV_FMonitor1,60s presented this way include determined consecutive long- and short-term variability indices for given data recorded by MONAKO system. Realization plan (Fig.1) will be feasible to create in declarative query language on the basis of the following continuous queries:

```
select
AVG(AGSE((FHR_AvgFMonitor1[1]-FHR_AvgFMonitor1[0]),NUMBER<24>,24))
as STV_FMonitor1 from
      select AGSE( AVG( AGSE((FHR1,FHR2,FHR3,FHR4),NUMBER<10>,4)),NUMBER<2>,1)
      as FHR_AvgFMonitor1 from InStream
      Filter InStream By ID_FMonitor = 1

select AVG( MAX(FHR_Avg24FMonitor1)-MIN(FHR_Avg24FMonitor1))
as LTV_FMonitor1 from
      select AGSE(AVG(AGSE((FHR1,FHR2,FHR3,FHR4),NUMBER<10>,4))),NUMBER<24>,24)
      as FHR_Avg24FMonitor1 from InStream
      Filter InStream By ID_FMonitor = 1
```

4. CONCLUSIONS

Database management systems drawn up so far are not efficient enough in biomedical applications. Ongoing research on query languages based on data streams resulted in creating some new concepts - including ours. Stream Algebra and continuous declarative query language is the basis of designed data stream management system. Access to the recorded biomedical signals will be realized by the presented query language. The presented example of application enables implementation of algorithm determining long and short term variability indices LTV and STV on the basis of declarative query language. A system realizing the techniques described in this paper is being developed at Institute of Medical Technology and Equipment.

Our solution enables to define query based on data streams that makes the updated answers currently available. Distinct from presented so far solutions, we do not enable – at present – linking of recorded in data streams information with the information stored in relational database. It is also worth noting that in the presented declarative query language the WHERE clause does not occur. Similar but very limited function is played by the FILTER BY clause intended exclusively to determine the conditions of data filtering. Presented declarative query language required drawing up and presenting assumptions of data algebra intended to analyses and record of biomedical signals. The ongoing researches are carried out in the framework on MONAKO system project. At present its architecture is centralized. However, applying database management system that carries out its tasks on the basis of presented assumptions enables the construction of monitoring system of distributed architecture

ACKNOWLEDGEMENT

This study was supported by the State Committee for Scientific Research, Warsaw, Poland (KBN Grant No. 4 T11E 006 22).

BIBLIOGRAPHY

1. JEŻEWSKI J. WRÓBEL J. HOROBA K. GRACZYK S. GACEK A. SIKORA J. Computerized perinatal database for retrospective qualitative assessment of cardiotocographic traces in „Current Perspectives in Healthcare Computing”. Editor: B. RICHARDS, BJHC Limited, pp.187-196. G. Britain, 1996
2. JEŻEWSKI J. WRÓBEL J. HOROBA K. Monitorowanie zagrożeń płodu wspomagane komputerem „Biocybernetyka i Inżynieria Biomedyczna 2000 – Tom 7. Systemy komputerowe i teleinformatyczne w służbie zdrowia” red. M. Nałęcz, Akademicka Oficyna Wydawnicza EXIT, pp. 97-119, Warszawa, 2001
3. WIDERA M. WINIARCZYK R. JEŻEWSKI J. GACEK A. WRÓBEL J. K. HOROBA: Strumienie danych w systemie monitorowania, analizy i klasyfikacji sygnałów biomedycznych, STUDIA INFORMATICA Volume 24, Number 3 (55), pp. 35-45, Gliwice 2003,
4. BABCOCK B., BABU S., DATAR M., MOTWANI R. WIDOM J. Models and Issues in Data Stream Systems, Invited paper in Proc.of the 2002 ACM Symp. on Principles of Database Systems (PODS 2002), June 2002
5. ARASU A. BABU S. WIDOM J. An Abstract Semantics and Concrete Language for Continuous Queries over Streams and Relations, Technical Report, November 2002.
6. ABADI D., CARNEY D. CETINTEMEL U. CHERNIACK M. CONVEY C. LEE S. STONEBRAKER M. TATBUL N. ZDONIK S. Aurora: A New Model and Architecture for Data Stream Management. In VLDB Journal, August 2003
7. MOTWANI R. WIDOM J. ARASU A. BABCOCK B. BABU S. DATAR M. MANKU G. OLSTON C., ROSENSTEIN J. VARMA R.. Query Processing, Resource Management, and Approximation in a Data Stream Management System In Proc. of the 2003 Conference on Innovative Data Systems Research (CIDR), January 2003
8. L. GOLAB, M.T. OZSU: Issues in data stream management, ACM SIGMOD Record Volume 32, Issue 2, pp.5-14, June 2003

Reference: J.Jeżewski, M.Widera, R.Winiarczyk, A.Matonia, T.Kupka, A.Gacek: Data stream processing in fetal monitoring system: II.Application to signal processing, Journal of Medical Informatics & Technologies, Vol. 6 ISSN 1642-6037, 2003, str.35-42