CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Online Research Database In Technology

Technical University of Denmark

A neural flow estimator

Jørgensen, Ivan Harald Holger; Bogason, Gudmundur; Bruun, Erik

Published in:
Proceedings of the IEEE Instrumentation and Measurement Technology Conference

Link to article, DOI:
10.1109/IMTC.1995.515299

Publication date:
1995

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Jørgensen, I. H. H., Bogason, G., & Bruun, E. (1995). A neural flow estimator. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference: Integrating Intelligent Instrumentation and Control (pp. 385-385). IEEE. DOI: 10.1109/IMTC.1995.515299

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Neural Flow Estimator

Ivan Harald Holger Jørgensen, Gudmundur Bogason and Erik Bruun

Electronics Institute
Building 349
Technical University of Denmark (DTU)
2800 Lyngby, Denmark
Phone (+45) 45 93 42 07Fax (+45) 42 88 01 17
E–Mail: ihhj@ei.dtu.dk , gudme@ei.dtu.dk and bruun@ei.dtu.dk

Abstract – This paper proposes a new way to estimate the flow in a micromechanical flow channel. A neural network is used to estimate the delay of random temperature fluctuations induced in a fluid. The design and implementation of a hardware efficient neural flow estimator is described. The system is implemented using switched–current technique and is capable of estimating flow in the µl/s range. The neural estimator is built around a multiplierless neural network, containing 96 synaptic weights which are updated using the LMS[1]–algorithm. An experimental chip has been designed that operates at 5V with a total current consumption of 2mA, resulting in a power consumption of 10mW. The dimensions of the chip core are 3 mm x 4.5 mm.

I. Introduction

Typically, deterministic signals have been used for heating the fluid in a flow channel in order to estimate the flow rate [1],[2], when using time of fleight measurements. The flow rate has then been based on some temperature measurement that are strongly dependent on the characteristics of the fluid. Therefore it has been necessary to have a relatively accurate model of the flow channel [1],[2].

In order to avoid the need for this detailed knowledge of the flow channel we propose a system, where we heat the fluid using a stochastic signal (White Noise) and then use a neural network to model the flow channel. The synaptic weights of the neural network reflect the delay in the flow channel, which can be translated to the flow rate.

A neural system has been designed which is capable of measuring the flow rate of fluids in a micro flow channel with very small geometrical dimensions. It is possible to measure flow rates in the microlitre per second range.

The neural network is designed using current mode and switched current techniques. This leads to a very compact system that has a very small power consumption.

A somewhat similar principle has been presented in [3, Chapter 11], where white noise is also used for heating a fluid. Then a computer is used for calculating the cross correlation between the measured temperature fluctuations at two different outputs. The delay is then determined by the maximum of this cross–correlation.

[1] Least Mean Square (LMS)

II. System Overview

A. Flow Channel

The flow estimator described in this paper is expected to operate with a flow sensor as described in [1]. The flow channel is schematically shown in Fig. 1.

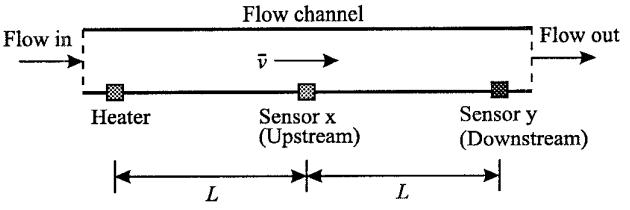

Fig. 1. Flow Channel

To estimate the flow rate of a fluid in the micro flow channel the fluid entering the channel is heated with white noise, shown as the "Heater" in Fig. 1. This generates some random temperature variations in the fluid. The flow of the fluid causes the temperature variations to propagate down the channel and the temperature variations are measured at two different locations in the flow channel, shown as "Sensor x" and "Sensor y" in Fig. 1.

Between the two points where the temperature is measured the flow channel can be modeled as a (lossy) transmission line acting on the temperature variations in the fluid [1].

The flow rate is found by estimating the delay in the transmission line which is approximately the time where the impulse response sequence of the line has its maximum peak.

The heating of the fluid and the measuring of the temperature in the flow channel is achieved using diodes integrated in the flow channel [1].

0-7803-2615-6/95/$4.00 © 1995 IEEE

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on February 16,2010 at 09:05:48 EST from IEEE Xplore. Restrictions apply.

B. Neural Estimator

The neural estimator is a discrete time system operating on temperature samples taken from the two measurement points in the flow channel (Sensor x and Sensor y). To facilitate this the temperatures fluctuations are sampled and held (S/H) as shown in Fig. 2.

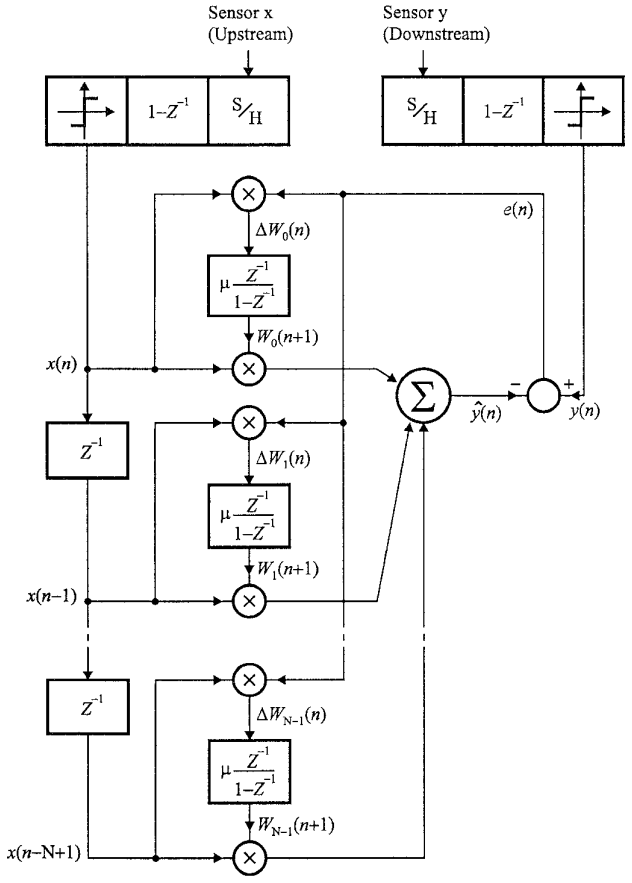

Fig. 2. Neural flow estimator.

The heating of the fluid with white noise will cause the temperature measured at the two locations down the flow channel to fluctuate around a mean value determined by the ambient temperature and by the power of the white noise used to heat the fluid.

The mean temperature of the heated fluid is removed by taking the derivative of the sampled temperature. This is done by subtracting the previous temperature sample from the actual temperature sample $(1-Z^{-1})$. This is effectively a highpass filtering of the measured temperature. Because this operation is performed at both measurement points, it has no effect on the operation of the neural estimator.

By digitizing the derivative of the sampled temperature into two levels (i.e. a binary signal) we introduce a very strong nonlinearity into the measured temperature. This wi increase the variance of the synaptic weights in the neur estimator which can be compensated for by decreasing tl update value μ, used by the LMS algorithm. The maj advantage associated with the digitization of the measure temperature is that the hardware complexity of the neur system is reduced considerably [4].

The neural estimator is configured to model tl transmission line of the flow channel between the tv measurement points. This is done by feeding temperatu samples taken from measurement point "Sensor x" into tapped delay line with 96 taps. Each of these taps is the multiplied by a synaptic weight and the sum of the multiplications is formed as shown in Fig. 2.

Because we have digitized the temperature samples, tl tapped delay can be implemented as a digital shift regist and all of the synaptic multiplications are replaced by eith an addition of a synaptic weight or not.

The learning sequence for the estimator tends to adjust tl synaptic weights of the neural network in such a way that tl error $e(n)$, between the summed output $\hat{y}(n)$, and tl temperature measured at point "Sensor y" $y(n)$, minimized in the least square sense.

The effect of minimizing this error, is that the synapt weights will eventually contain an estimate of the impul response sequence of the flow channel between the tv measurement points "Sensor x" and "Sensor y". Tl resolution of this estimate is determined by the number synaptic weights, in our case it is 96.

The delay of the thermal fluctuations between the tv measurement points is given by the peak of the impul response sequence i.e. it can be found as the numerical largest synaptic weight in the neural estimator. The resolutic of the estimated delay is therefore determined by the tin between two taps in the tapped delayline which in turn determined by the sampling frequency.

Because we only need to identify the largest synapt weight, the learning of the neural estimator does not have proceed until the final values of the weights have be reached. Hence, we obtain a very fast response time from tl neural estimator.

III. Simulations

A simulation illustrating the response time of the neur estimator is shown in Fig. 3.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on February 16,2010 at 09:05:48 EST from IEEE Xplore. Restrictions apply.

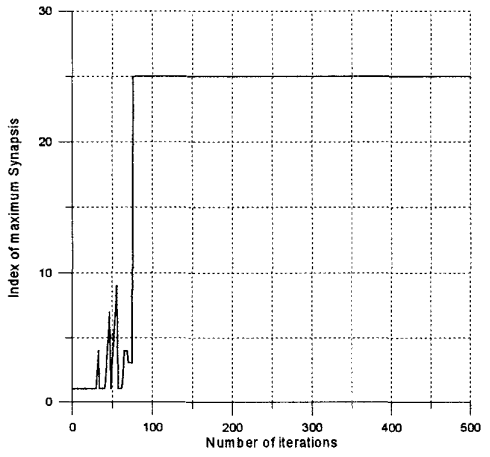

Fig. 3. Time history of the numerically largest synaptic weight, estimating the delay in the flow channel (Flow rate)

Fig. 3 shows the index of the numerically largest synaptic weight as a function of the number of iterations performed during learning. The simulation is based on a simplified model of the flow channel, and is only intended to illustrate the operation of the system. We are not interested in having specific knowledge about the flow channel, it is the job of the neural estimator to provide this information. The simulation is based on the parameters shown in Table 1.

Table 1. Parameters used for the simulation of the flow channel.

Parameters		Value
Sampling period	ΔT	*0.05s*
Distance between x and y	L	*1500 μm*
Flow rate	v	*1200 μm/s*
Scaling	μ	*0.001*

Before the simulation is started, all of the synapses are reset to zero. Then the neural estimator starts the sampling of the temperature in the flow channel and updates the synaptic weights. In Fig. 1, the distance from the Heater to Sensor x is 1500μm. With the flow rate shown in Table 1, this corresponds to a delay of 1.25s or 25 samples. This can be seen on Fig. 3 as the flat region from iteration 0 to 25. At this point, samples are filled into the tapped delay at Sensor x, which results in the fluctuations of the synapses with the lowest index i.e. closest to Sensor x. This will continue for iteration 25 to 50. From iteration 50, Sensor y also gets samples from the flow channel. Hereafter the neural estimator begins to adjust its synaptic weights, in order to minimize the error $e(n)$. From Fig. 3 we see that it only takes approximately 25 additional updates of the synaptic weights to get a stable maximum synaptic weight. We also see that the numerically largest synaptic weight has index 25, corresponding to a delay of $25 \cdot 0.05s = 1.25s$. The calculated delay is

$$\frac{1500\mu m}{1200\mu m/s} = 1.25s$$

which is the same as the estimated delay. The response time of the neural estimator, defined as the time from iteration 0 to a stable estimate, is 75 samples, corresponding to 3.75s.

In Fig. 4 we have a simulation showing the synaptic weights after 4096 iterations. This figure shows the estimated impulse response of the flow channel between the measurement points at Sensor x and Sensor y.

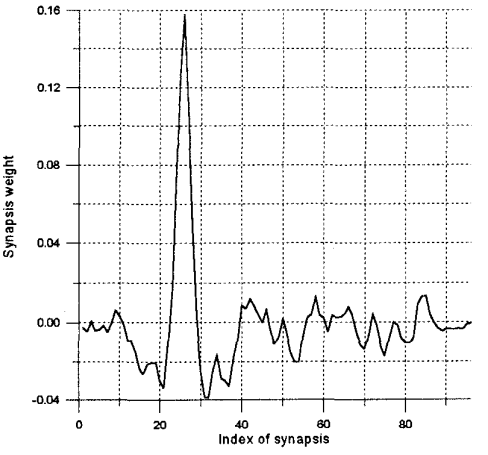

Fig. 4. Synaptic weights after 4096 iterations

The fluctuations of the synaptic weights is a result of the coarse quantization of the sampled temperature to a binary value.

Also, a simulation showing the evolution of the numerically largest synapsis (synapsis 25) as a function of the number of iterations is shown in Fig. 5.

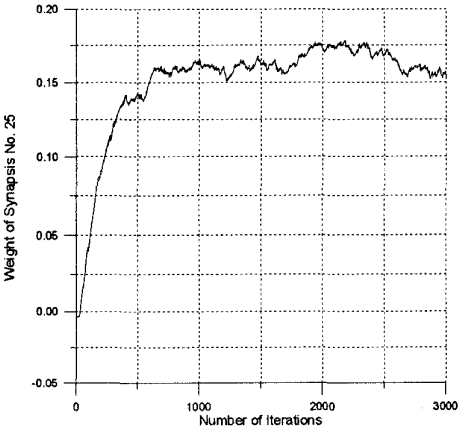

Fig. 5. Synaptic weight No. 25 as a function of time

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on February 16,2010 at 09:05:48 EST from IEEE Xplore. Restrictions apply.

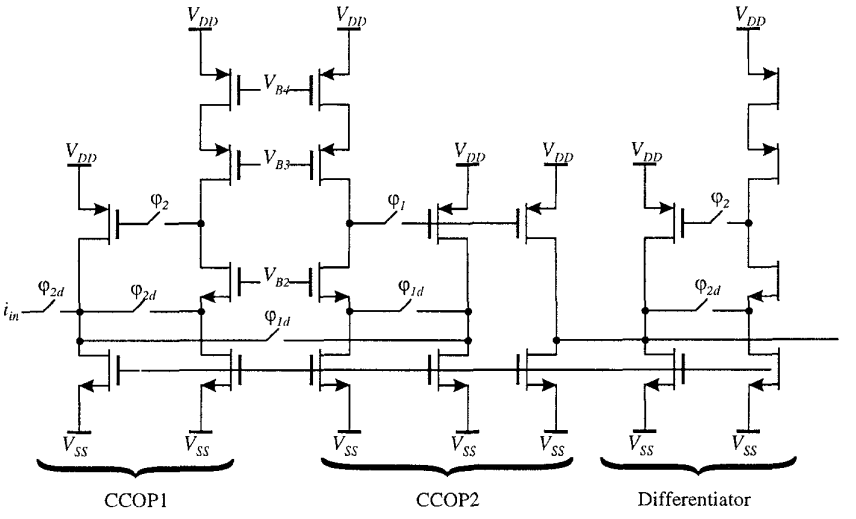

Fig. 6. Sample / Hold and Differentiator

Fig. 5 shows that it takes more than 500 iterations for the synaptic weight to stabilize.

IV. Circuit Description

As described in the previous section, the system consists of four basic building blocks: Sample and hold (S/H), differentiator, comparator and a neural estimator which is built from integrators.

All of the circuits are based on switched current techniques, using folded cascode Current Copiers (CCOP's) [5] that perform an inversion and a half clock period delay of the signal current ($-Z^{-1/2}$). The operation of the switched current circuits is based on a two phase clock scheme, where the clock phases are denoted φ_1 and φ_2.

Based on noise analysis, the bias currents used in the current copier cells was chosen to 5μA. The memory transistors, shown as transistors with a switch connected at the gate, were implemented using PMOS transistors instead of NMOS transistors. This gives a more regular layout. All transistors that have their gate tied to V_{B1} and V_{B4} are current sources and transistors with their gate tied to V_{B2} and V_{B3} are cascode transistors.

In sections A,B and C, we present a brief description of the building blocks used for implementing the neural estimator.

A. Sample/Hold and Differentiator

In Fig. 6 the circuit implementing the Sample/Hold and differentiator is shown.

The first two current copiers CCOP1 and CCOP2 perform the Sample and Hold function. CCOP1 samples the signal at clock phase two and this sample is the fed into CCOP2 on clock phase one. The signal held in CCOP2 is fed to differentiator.

B. Comparator

The Comparator, shown in Fig. 7, is built around differential pair.

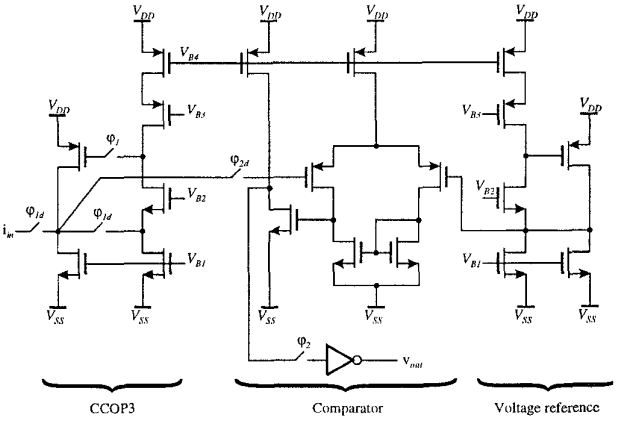

Fig. 7. Comparator

The input signal is first clocked into the current copie CCOP3 at clock phase one. At clock phase two the curren stored in the current copier CCOP3 is sent to the gate of on of the transistors in the differential pair. This is a very hig impedance node, and this will cause the voltage to swin either to the positive or negative side, depending on the sig of the current stored in CCOP3. The differential pair com pares this voltage with a reference voltage supplied by voltage reference and delivers a binary output.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on February 16,2010 at 09:05:48 EST from IEEE Xplore. Restrictions apply.

C. Synaptic weight (Integrator)

Fig. 8 shows one of the synaptic weights. The synaptic weight consists of a programmable current divider making the scaling factor μ, and a switched current integrator. The output current of the integrator corresponds to the synaptic weight.

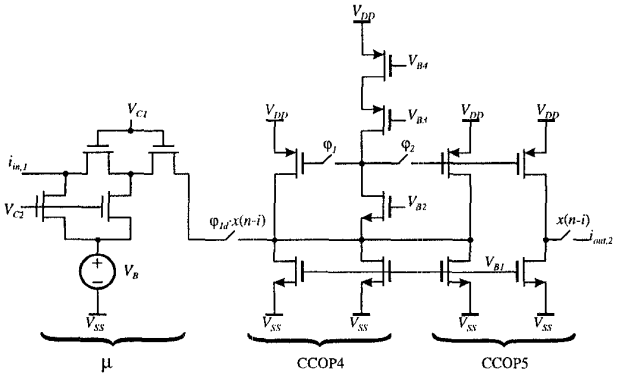

Fig. 8. The ith Synaptic weight H(z) (Integrator)

By cascading two current copiers (CCOP4 and CCOP5) we get a delay line. And by feeding the output of this delay line back to its input the accumulator (Integrator) is formed. The switches at the input and output of the integrator are the multipliers shown in Fig. 2. Because the temperature samples are digitized to a binary value, the multiplications are replaced by switches. These switches are controlled by the digitized samples in the tapped delay.

For the LMS algorithm to converge, the scaling factor μ must be very small. Simulations have shown that μ should be smaller than 0.01, and preferably 0.001 to keep the variance of the synaptic weights reasonably low during learning. Simulations also showed that the linearity of the scaling factor μ was of no importance for the learning of the neural estimator. In order to get such a small scaling factor it is not feasible to use current mirrors. Therefore we have used a simple resistive scaling based on MOS transistors operating in their linear region (Resistors). By controling the gate voltages V_{C1} and V_{C2}, the scaling factor μ can be programmed. The voltage source V_B is used for canceling any DC–currents flowing to the integrator.

V. Experimental Results

An experimental chip with 96 synapses has been designed in an industry standard 2.4μm CMOS process. The dimensions of the chip core are 3 mm x 4.5 mm and the number of transistors is approximately 7500. The chip is designed to operate at 5V with a total current consumption of 2mA resulting in a power consumption of 10mW.

No measurement results are available at the moment because the chip is still being processed. Hopefully, some measurement results are available at the presentation of this paper.

VI. Conclusion

In this paper we have presented a neural estimator intended for estimation of the flow rate in micromechanical flow channels. The estimate is based on the relationship between the shape of an impulse response and the delay caused by a filter with that impulse response, the filter being the flow channel seen between two measurement points. The accuracy of the flow estimator is not known at the moment, as we have not yet tested the circuit.

We have shown that it is feasible to implement a multiplierless neural estimator with 96 synapses in switched current technique, with small power consumption and a small chip area.

References

[1] C. Yang and H. Søberg, "Monolithic flow sensor for measuring millilitre per minute liquid flow", *Sensors and Actuators A*, Vol. 33, pp. 143-153, 1992.

[2] T. S. J. Lammerink, N. R. Tass, M. Elwenspoek and J. H. J. Fluitman, "Micro–liquid flow sensor", *Sensors and Actuators A*, Vol. 37–38, pp. 45-50, 1993.

[3] W. Göpel, J. Hesse and N. Zemez, "Sensors A Comprehensive Survey", VCH Verlagsgesellschaft mbH, Mechanical Senors, Vol. 7,1994

[4] H. Qiuting and G. S. Moschytz, "Analog Multiplierless LMS Adaptive Filter Structures" , *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, Vol. 40, No. 12, pp. 790–794, December 1993.

[5] C. Toumazou, J. B. Hughes and N. C. Battersby, "SWITCHED CURRENTS an analog technique for digital technology", Peter Peregrinus Ltd., London, United Kingdom, 1993.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on February 16,2010 at 09:05:48 EST from IEEE Xplore. Restrictions apply.