Proceedings of the 21st International Conference on Global Research and Education (Inter-Academia 2024)

Echo State Network-Based Estimation of Photoplethysmography Sensor-To-Skin Contact Force

M. Szumilas* and M. Wielemborek

Faculty of Mechatronics, Institute of Metrology and Biomedical Engineering, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland

Doi: 10.12693/APhysPolA.146.369 *e-mail: mateusz.szumilas@pw.edu.pl

A photoplethysmographic signal, widely used in cardiovascular monitoring, is susceptible to the sensor's mounting conditions, including the contact force at the sensor-to-skin interface. We aimed to extract this concomitant parameter from a reflective photoplethysmographic signal to enable better observation of varying measurement conditions. Evaluation of a regressor based on an echo state network yields promising results when modeling the relationship between a reference force signal delivered from a force-sensitive resistor and the infrared and red photoplethysmographic signal components with an average normalized root mean square error of 0.101 (range of 0.051–0.150) for the considered test cases. The echo state network regressors using as few as 10 neurons show potential for deployment and online adaptation in resource-constrained hardware, e.g., microcontrollers.

topics: echo state networks (ESN), photoplethysmography (PPG), sensor-to-skin contact force

1. Introduction

Photoplethysmography (PPG), an optical technique for measuring blood volume changes in tissues, is widely used in cardiovascular monitoring [1]. PPG sensors consist of a light source, which irradiates the skin, and a photodetector, which measures the light that becomes reflected or transmitted (depending on the photodetector's position relative to the light source). The received signal may serve, e.g., for indirect estimation of blood flow, blood oxygen saturation, or blood pressure [2]. However, this signal remains susceptible to the sensor's mounting conditions, including the temperature and the contact force at the sensor-to-skin interface [3]. Changes in the latter contribute to baseline wandering in the raw photodiode signal and affect some of the morphological features in the PPG signal [4, 5]. The mentioned changes in raw signal are considered a disturbance and commonly attenuated via high-pass filtering. In contrast to this approach, we propose to process the raw signal and extract the contact force changes. The primary goal is to improve PPG-based monitoring by allowing for differentiation between changes due to the mounting conditions and those resulting from blood flow alterations of cardiovascular origin. This would make it possible, for example, to identify signal deterioration related to external pressure in bed-confined patients who may accidentally lean on the sensor.

2. Materials and methods

The measurement module consists of a mounting plate made using 3D printing technology from *polylactic acid* (PLA) material, Analog Devices MAX30102 reflective pulse oximeter & heart-rate sensor (PPG sensor), DFRobot RP-C7.6-ST thin-film *force-sensitive resistor* (FSR), and Redox S90 MG micro servo (Fig. 1). The device is attached to subject's body using non-elastic velcro straps.

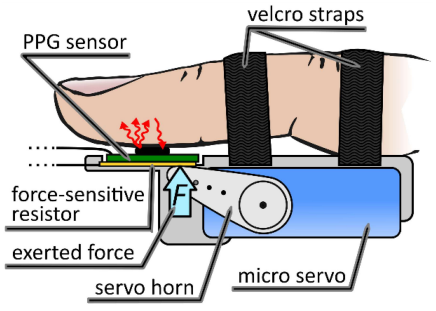

Fig. 1. Schematic of the study setup with the measurement module attached to a finger. The servo modulates the sensor-to-skin contact force $\boldsymbol{F}$, while signals are collected from the *photoplethysmographic* (PPG) sensor and the force-sensitive resistor.

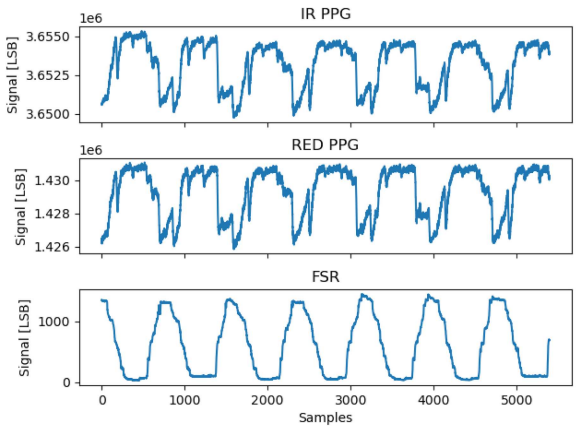

Fig. 2. Example of acquired raw signals.

The PPG sensor positioning is achieved by loose mating with a corresponding cutout in the module. The servo horn pushes the PPG sensor against the skin with the FSR between the sensor and the horn. The force that the PPG sensor exerts on the body is modulated by the oscillatory movements of the servo horn. The system is controlled using the STM32 NUCLEO-G031K8T6 board. Data from the PPG and force sensors are acquired concurrently at 200 samples/sec rate and streamed to the host computer via serial interface.

The measurements were conducted on the thumbs of two subjects, who were members of the research team. For each subject, data were collected under nine different test conditions, including three repositionings of the module on the finger and three force modulation rates, f_{mod} (0.25, 1, and 4 Hz). Changing the positioning of the finger involved manual adjustment of the initial pressure of the finger against the PPG sensor using the non-elastic strap with simultaneous visual verification of the recorded signal quality (with an aim to make the systolic peak of the pulse wave visible and to avoid clipping of the PPG and FSR signals). Figure 2 presents an example of seven cycles of raw PPG signals (acquired using red and infrared (IR) light) and the corresponding raw FSR signal.

To achieve transformation between the FSR and PPG signals, we employ a leaky *echo state network* (ESN), a machine learning model that belongs to a broader field of *reservoir computing* (RC) [6, 7]. In the RC approach, the input data is used to excite a system (called a "reservoir") capable of developing potentially complex and non-linear dynamics. As the dimensionality of the data represented with the reservoir states is increased compared to its input, it facilitates obtaining the desired input vs. target relationship with relatively simple methods such as linear regression. It is worth noting that the reservoir is a fixed entity, i.e., it does not undergo training-based adjustments, and its parameters are set only during initialization. The ESN itself is implemented as a *recurrent neural network* (RNN) with a randomly initialized hidden layer (typically having a sparse set of recurrent connections) that serves as a reservoir. An adequate adjustment of its hyperparameters makes the network gain an echo state property. It means that the fading memory of the network input uniquely defines the reservoir's state, where it becomes embedded. The only network element that undergoes supervised learning is the output layer.

The ESN state is stored in the reservoir vector $\boldsymbol{x}$ With each new data sample at the network's input, its state becomes updated according to

$$\boldsymbol{x}(t) = (1-\alpha)\,\boldsymbol{x}(t-1) + \alpha \tanh\Big(W_{in}\,[1\,;\,\boldsymbol{u}(t)] + W\,\boldsymbol{x}(t-1)\Big), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^N$ is a vector of reservoir activations at the time step t, N is the number of neurons in the reservoir, $\boldsymbol{u}(t) \in \mathbb{R}^M$ is an M dimensional input signal at the time step t, $W_{in} \in \mathbb{R}^N \times \mathbb{R}^{M+1}$ and $W \in \mathbb{R}^N \times \mathbb{R}^N$ are the input and reservoir weight matrices, respectively, $[\cdot\,;\,\cdot]$ is a vertical concatenation operator, $\alpha \in (0, 1]$ is the leaking rate, and $\tanh(\cdot)$ is the element-wise hyperbolic tangent function.

The ESN output is defined as

$$\boldsymbol{y}\,(t) = W_{out}\,[1\,;\,\boldsymbol{x}\,(t)\,], \tag{2}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^P$ is the P-dimensional network output and $W_{out} \in \mathbb{R}^P \times \mathbb{R}^{N+1}$ is the output weight matrix.

The reservoir matrix W and input weights W_{in} are initialized randomly from uniform distributions over $[-0.5; 0.5]$ and $[-0.5\omega; 0.5\omega]$, respectively, where ω is the input scaling parameter. The matrix W is typically sparse, with sparsity (i.e., the fraction of non-zero elements in the matrix) denoted by s. An important ESN hyperparameter is the spectral radius $\rho(W)$ of the reservoir, i.e., the maximum absolute eigenvalue of W. Considering a network with leaky integration in nodes ($\alpha \leq 1$), the effective spectral radius is calculated according to [8] via the equation

$$\rho\left(\tilde{W}\right) = \rho\big(\alpha W + (1-\alpha\, I)\,\big), \tag{3}$$

where $I \in \mathbb{R}^N \times \mathbb{R}^N$ is an identity matrix. In most cases, setting $\rho(\tilde{W}) < 1$ is sufficient to ensure the echo state property, however, a reservoir-specific adjustment may be necessary. Usually, the spectral radius of the matrix is adjusted in two steps: the matrix is element-wise divided by its current spectral radius and then element-wise multiplied by the value of the desired spectral radius.

As the ESN state is a function of its past inputs, the network has to process a sequence of input initialization data before its output may be considered valid. Therefore, we compute the length of an initialization data, p_{init}, which allows the uninitialized

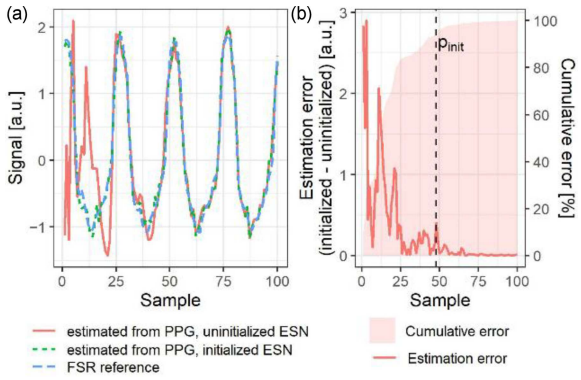

Fig. 3. Illustration of p_{init}: (a) FSR signal estimation from uninitialized vs initialized ESN and (b) corresponding absolute estimation error and the p_{init} index position ($N_{\text{init}} = 100$, $\beta = 0.95$).

network's output $g_{\text{uninit}}(x)$ to converge with the output of a pre-initialized network $g_{\text{init}}(x)$. The estimation error e_i for the data point x_i is defined as

$$e_i = \text{abs}\left(g_{\text{init}}\left(x_i\right) - g_{\text{uninit}}\left(x_i\right)\right). \tag{4}$$

The length p_{init} is equal to the index of the point x_i at which the cumulative absolute error becomes greater than a predefined fraction β of the total cumulative error of the analyzed data (Fig. 3 and the following equation), i.e.,

$$p_{\text{init}} = \underset{p}{\text{argmin}} \left(\sum_{i=0}^{p} e_i \geq \beta \sum_{i=0}^{N_{\text{init}}} e_i \right), \tag{5}$$

where N_{init} is the length of the analyzed data, β should be assigned from the $(0;1)$ range.

The choice of the ESN hyperparameters determines the network's performance. In the presented solution, their values are tuned using *Bayesian optimization* (BO) [9]. The core concept of the BO approach is to establish a probabilistic model of an optimized function $f(x)$ on a bounded set A and exploit this model to make an informed decision (i.e., a decision based on the previous $f(x)$ evaluations) about which new point $x \in A$ is expected to yield the most significant increase in the function value and is worth evaluating. This is particularly suitable for optimizing functions that are expensive to evaluate (e.g., it is time-consuming or it spends other valuable resources), which would not be feasible using exhaustive optimization methods, e.g., grid search.

After normalization and downsampling to 100 samples/sec, measurements from the PPG calibration sequence are fed into the network. The first 300 points from the sequence initialize the ESN internal state, and the corresponding network activations are dropped. The following 1600 (for 0.25 Hz modulation rate) or 800 (for 1 Hz and 4 Hz modulation rates) samples are split into the train and validation

TABLE I

Network hyperparameters subject to optimization.

Symbol	Description	Range
N	number of neurons in the reservoir	10–300
α	network leaking rate	0.01–0.99
s	sparsity of W	0.02–0.9
ω	input scaling of W_{in}	0.01–10
ρ	spectral radius of W	0.5–1.5
t_{lag}	estimation delay (in samples)	0–10

datasets in a 3:1 ratio. The reservoir activations are collected and stored in the activation matrix S. A linear regression model is then fitted on the training set to calculate the W_{out} and establish the relationship between the readouts from the FSR sensor and S with a configurable delay of output vs input by t_{lag} (to compensate for the possible delay between the force change and the corresponding raw PPG response). This yields the signal estimation equation

$$\widehat{\mathbf{FSR}}\left(t - t_{\text{lag}}\right) = W_{out}\left[1\,;\,\boldsymbol{x}\left(t\right)\right]. \tag{6}$$

The network is optimized over the hyperparameters given in Table I.

For each set of hyperparameters, a pool of five ESNs is evaluated to account for their inherent randomness. The network's performance metric is the *normalized root mean square error* (NRMSE), calculated with the validation data. After training all the ESNs in the pool, the within-pool minimum value of NRMSE becomes the score for the specific set of hyperparameters. This score is then minimized using the BO, with the expected improvement (EI) as the acquisition function ($\xi = 0.01$) and the Matérn 5/2 kernel. The process is initialized with 20 points; after that, $f(x)$ is evaluated 40 times. For the ESN with the lowest NRMSE in the pool, p_{init} is calculated independently in 3 non-overlapping segments of the combined training and validation data ($N_{\text{init}} = 300$, $\beta = 0.95$), and then averaged.

For each measurement sequence, the estimation error of the best-performing ESN (i.e., the one with the lowest NRMSE during BO) is evaluated on all the other sequences. Each time the network is initialized with $2p_{\text{init}}$ data points.

The data processing and subsequent ESN training and evaluation are performed in the R environment ver. 4.3.2 [10].

3. Results

The parameters of the best-performing models are summarized in Table II and Fig. 4.

TABLE II

Parameters of the best-performing models. The normalized root mean square error (NRMSE) is evaluated for the validation data ($e_{valid.}$) and externally, as an average from all the modified measurement setups (e_{extern}).

Subject	Finger placement	f_{mod} [Hz]	$e_{\mathrm{valid.}}$	$\overline{e_{\mathrm{extern}}}$	$\overline{p_{\mathrm{init}}}$	t_{lag}	N	α	s	ω	ρ
1	I	0.25	0.150	0.476	64	10	300	0.31	0.71	0.98	1.23
		1	0.064	0.615	24	7	281	0.93	0.86	0.19	1.03
		4	0.065	0.310	40	6	10	0.40	0.16	0.01	1.27
	II	0.25	0.094	1.049	19	8	167	0.69	0.44	0.01	1.03
		1	0.099	0.365	6	10	10	0.48	0.18	4.20	0.79
		4	0.139	0.661	27	0	38	0.21	0.43	0.26	1.06
	III	0.25	0.121	0.250	41	4	13	0.15	0.80	3.02	1.11
		1	0.129	0.505	13	2	71	0.34	0.29	1.86	0.87
		4	0.111	0.298	17	9	179	0.77	0.38	1.70	1.25
2	I	0.25	0.133	0.324	1	9	25	0.97	0.77	8.51	0.94
		1	0.103	0.996	162	3	51	0.12	0.57	0.55	1.44
		4	0.127	0.349	63	8	250	0.90	0.80	0.02	1.19
	II	0.25	0.073	0.330	157	10	10	0.03	0.11	2.52	1.5
		1	0.080	0.383	36	0	136	0.78	0.07	0.65	1.12
		4	0.089	0.493	13	6	221	0.62	0.80	0.96	0.82
	III	0.25	0.051	0.930	127	2	55	0.24	0.67	0.46	1.34
		1	0.087	0.264	32	10	10	0.24	0.79	1.22	0.89
		4	0.100	0.414	24	3	125	0.77	0.63	0.04	1.04

The NRMSE errors of the models for the considered test cases ($e_{\mathrm{valid.}}$) are in the range of 0.051–0.150, with an average of 0.101. Each ESN model with the lowest optimization-stage error was tested externally on the data from the remaining measurement setups to verify its prediction quality under changing conditions, yielding an average $\overline{e_{\mathrm{extern}}}$ error across all the models equal to 0.501 (range of 0.250–1.049).

Considering the estimated $\overline{p_{\mathrm{init}}}$ and the data sampling rate (100 samples/sec after downsampling), all the tested networks should adapt to significant signal changes within 2 s.

4. Discussion

The model characteristics that have an essential impact on its usability are the reservoir size N and the number of samples required for initialization p_{init}. Models with fewer neurons (as low as 10 neurons in our case, as shown in Table II) require limited resources for operation and possible online retraining. The dynamics of model response to input signal changes might be partially predicted by estimating its startup time — models with high p_{init}, particularly when it noticeably exceeds the number of reservoir's neurons, will respond slowly with possible overshoot. Such insensitivity to the input is expected in networks with relatively low leaking rates (here, reaching 0.03).

Some model transferability issues after changing the measurement setup are visible in the results. Estimated NRMSEs exhibit relative increases in the range of 2.1–18.2 between the modified and the training setup ($\overline{e_{\mathrm{extern}}}/e_{\mathrm{valid.}}$ in Table II).

This behavior may be an effect of tuning the network to a signal with a particular modulation frequency, especially considering that the error increase is lowest when tested on setups with unchanged f_{mod} (mainly for 1 Hz and 4 Hz), even after changing the subject (Fig. 4). It is worth considering replacing the force modulation pattern with, e.g., frequency sweep to mitigate this effect.

Due to the relatively short period of acquired data, we did not gain insight into the model's response to long-term signal drift. In the case of using a measurement module equipped not only with a PPG sensor but also featuring a force-modulation capability, periodic retraining of the ESN should be possible to avoid model quality loss.

Given the risk of overfitting the data from a specific measurement setup, two different model optimization paths are viable, depending on the desired application. The first option is to aim for better generalisability to improve the direct estimation of contact force in the absence of signals other than raw PPG. This requires modifying the force modulation pattern to diversify the network excitations and states for both model training and validation. The second contrasting option is to take advantage of the network's sensitivity to the measurement configuration. This effect allows for the implementation

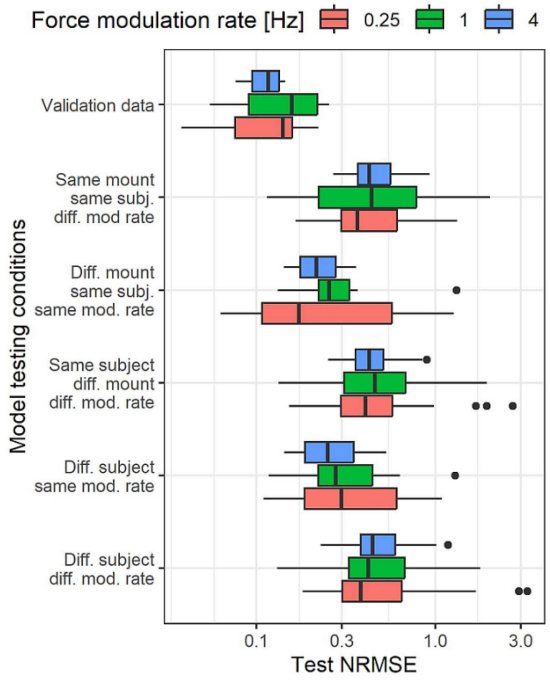

Fig. 4. The summary of NRMSE scores achieved by the best-performing ESN models when tested on the data from different PPG sensor setups.

of an anomaly detector that can be used for monitoring not explicitly the sensor's contact force but rather the overall stability of its working conditions, contributing to the field of PPG signal quality control [11].

As the literature review showed no comparable studies aiming to retrieve sensor contact force from the PPG data, we consider this approach worth further investigation, especially by acquiring more training data covering various measurement conditions and fine-tuning the optimization target functions.

5. Conclusions

The contact force at the sensor-to-skin interface of a PPG sensor can be retrieved from the raw signal using relatively small recurrent network-based models (with as few as 10 neurons) suitable for resource-constrained devices (e.g., microcontrollers) and optional online adaptation. For the considered test cases, an average NRMSE error of 0.101 (range of 0.051–0.150) is a promising result warranting further investigation.

References

[1] J. Park, H.S. Seok, S.-S. Kim, H. Shin, *Front. Physiol.* **12**, 808451 (2022).

[2] G. Chen, L. Zou, Z. Ji, *APL Bioeng.* **8**, 031501 (2024).

[3] I. Pi, I. Pi, W. Wu, *SN Appl. Sci.* **4**, 21 (2022).

[4] J.M. May, E. Mejía-Mejía, M. Nomoni, K. Budidha, C. Choi, P.A. Kyriacou, *Sensors* **21**, 8421 (2021).

[5] A. Chandrasekhar, M. Yavarimanesh, K. Natarajan, J.-O. Hahn, R. Mukkamala, *IEEE Trans. Biomed. Eng.* **67**, 3134 (2020).

[6] H. Jaeger, "The "Echo State" Approach to Analysing and Training Recurrent Neural Networks — with an Erratum Note", Technical Report no. 148, German National Research Center for Information Technology, 2001.

[7] M. Lukoševičius, in: *Neural Networks: Tricks of the Trade*, Vol. 7700, Eds. G. Montavon, G.B. Orr, K.-R. Müller, 2012 p. 659.

[8] H. Jaeger, M. Lukoševičius, D. Popovici, U. Siewert, *Neural Netw.* **20**, 335 (2007).

[9] J. Snoek, H. Larochelle, R.P. Adams, in: *Advances in Neural Information Processing Systems*, Vol. 25, Eds. F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger, 2012.

[10] R Core Team, R: A Language and Environment for Statistical Computing, 2024.

[11] T. Desquins, F. Bousefsaf, A. Pruski, C. Maaoui, *Appl. Sci.* **12**, 9582 (2022).