Reconstruction of ECG signals using RLS and ADAM optimizer

Elvis Rodas and Henrik Thunqvist

***Abstract*—8 patients have measured ECG-signals. The last 30 seconds of a 10 minute long signal is missing for each one. To reconstruct the last part, two methods have been used; A recursive least-squares (RLS) algorithm and the ADAM optimizer.The ADAM optimizer was proven to be computational effective.**

I. Introduction

The project is based on the 2010 challenge published by the online forum PhysioNet [1]. Every year the challenge focus on a different issue in the field of biomedicine. The 2010 challenge is about reconstructing a missing electrocardiogram (ECG) signal from different patients. In order to estimate the lost part of the signal, two different correlated signals are available. An example of these signals for patient number two is shown in Fig. 1. The goal is to estimate the last 30 seconds of the target signal using the two correlated signals x_1 and x_2.

Assuming that the target signal can be written as linear combination of the correlated signals x_1 and x_2, it can be modelled as

$$x_T[n] = \sum_{i=0}^{N} a_i x_1[n-i] + \sum_{i=0}^{M} b_i x_2[n-i] + w[n] \tag{1}$$

where a_i and b_i are the filter taps for the signals x_1 and x_2 respectively and $w[n]$ is unknown disturbance. N and M are the filter length which tells how many recent input values is used for the estimation.

The real missing part is also provided to evaluated the reconstruction performance of the chosen methods. The quality functions implemented for testing the performance are defined as

$$Q_1 = 1 - \frac{\text{mse}(x_T[n], \hat{x}_T)}{\text{var}(\hat{x}_T[n])} \tag{2}$$

$$Q_2 = \frac{\text{cov}(x_T[n], \hat{x}_T[n]))}{\sqrt{\text{var}(x_T[n])\text{var}(\hat{x}_T[n])}} \tag{3}$$

where *mse* denotes the mean square error, *var* the variance and *cov* the covariance of the variables in their arguments.

II. Filter Methods

The methods used for reconstruction of the missing signal $\hat{x}_T[n]$ are the Recursive Least Square (RLS) and the Adaptive Moment Estimation (ADAM) optimizer.

Before the signals goes through both filters, they are made to have zero mean. This because colinearity is reduced which makes the coefficients less sensitive to small changes in model and we receive better estimates. The subtracted mean value is added back to the reconstruction.

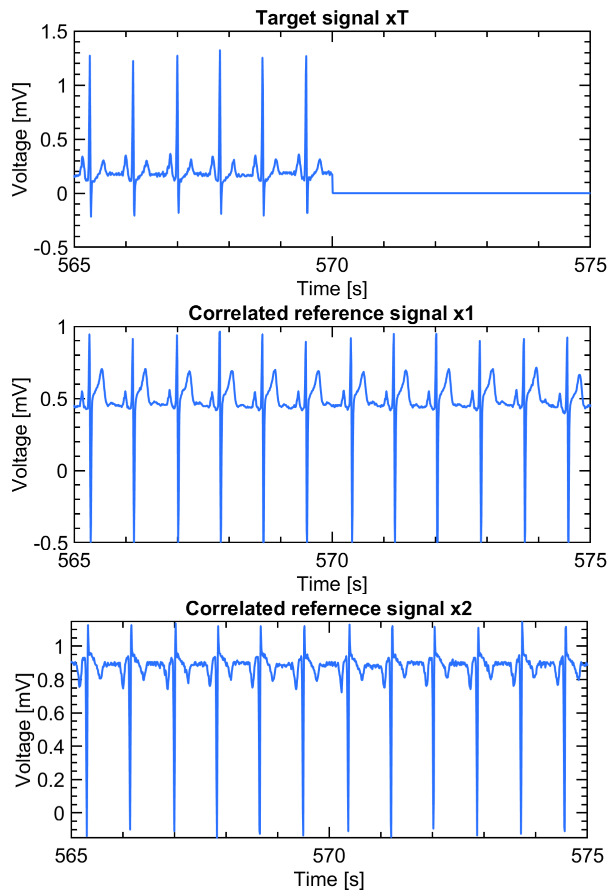

Fig. 1. Signals for patient number two. The missing part of the target signal are set to zero.

A. RLS algorithm

The Recursive Least Square (RLS) is an adaptive filter method. The aim is to minimize the cost function C defined as the weighted least squares error as

$$C(\boldsymbol{w}_n) = \sum_{k=0}^{n} \lambda^{n-k} e[k]^2, \tag{4}$$

where the weight $0 < \lambda < 1$ is the forgetting factor. It tells how much of the past signals we want to use in the new estimation.

The error signal $e[n]$ and the desired signal $d[n]$ are defined in the negative feedback diagram shown in Fig. 2. The filter coefficients $h[n]$ are updated for a new data $x[n]$ arrives.

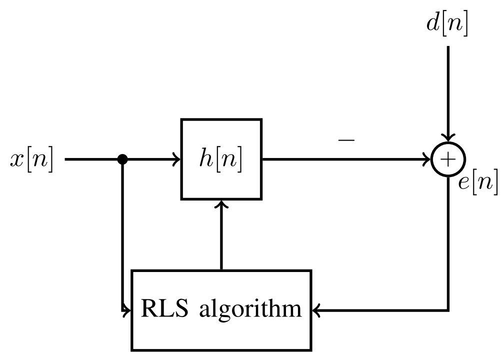

Fig. 2. RLS diagram.

From the diagram the error can be written as,

$$e[n] = d[n] - \sum_{k=0}^{M-1} h[k]x[n-k] \tag{5}$$

which in algebraic notation can be written as,

$$e = d - Ah \tag{6}$$

where

$$e = \begin{bmatrix} e[0] & e[1] & \cdots & e[N-1] \end{bmatrix}^T \tag{7}$$

$$d = \begin{bmatrix} d[0] & d[1] & \cdots & d[N-1] \end{bmatrix}^T \tag{8}$$

$$A = \begin{bmatrix} x[0] & x[-1] & \cdots & x[-M] \\ x[1] & x[0] & \cdots & x[1-M] \\ \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[N-2] & \cdots & x[N-M] \end{bmatrix} \tag{9}$$

$$h = \begin{bmatrix} h[0] & h[1] & \cdots & h[M-1] \end{bmatrix}^T \tag{10}$$

The L_2 norm of the error is

$$\begin{aligned} |||e||_2^2 &= ||d - Ah||_2^2 && (11)\\ &= (d - Ah)^T(d - Ah) && (12)\\ &= d^Td - h^TA^Td - d^TAh + h^TA^TA^Th && (13) \end{aligned}$$

By taking the gradient with respect to the filter taps h we get

$$A^TAh^* = A^Td \tag{14}$$

where h^* is the optimal filter tap. It can be written as

$$h^* = (A^TA)^{-1}A^Td \tag{15}$$

A problem rises when trying to inverse the matrix $(A^TA)^{-1}$. The more the data the more expensive the matrix inversion. A solution to this problem is by utilizing the Matrix inversion lemma and a recursive LS (RLS).

Lema 1 (Matrix inversion lemma)**.** *Let A, B, C and D be positive definite matrices of dimentions $M \times M$, $M \times N$ and $N \times N$ respectively. If,*

$$A = B^{-1} + CD^{-1}C^T, \tag{16}$$

then,

$$A^{-1} = B - BC(D - C^TBC)^{-1}C^TB. \tag{17}$$

Proof. The proof can be found in [2]. □

Let u_i denote the most N recent input defined as

$$u_i = \begin{bmatrix} x[i] & x[i-1] & \dots & x[i-(M-1)]^T \end{bmatrix} \tag{18}$$

The equation for optimal filter tap in can be written as,

$$\phi_n h_{n-1}^* = Z_n \tag{19}$$

where,

$$\phi_n = \sum \lambda^{(n-1)-i} u_i u_i^T, \tag{20}$$

$$Z_n = \sum \lambda^{(n-1)-i} u_i.d[i] \tag{21}$$

Equation (19) is the same as (14) with the difference that the forgetting factor λ defined in (29) was added. The optimal filter taps are computed by

$$h^* = \phi_n^{-1} Z_n \tag{22}$$

which is the equation (15) with the forgetting factor added.

Equations (20) and (21) can be written in terms of the their previous values ϕ_{n-1} and Z_{n-1} respectively as

$$\phi_n = \lambda\phi_{n-1} + u_n \mathbb{1} u_n^T \tag{23}$$

$$Z_n = \lambda Z_{n-1} + u_n d[n] \tag{24}$$

the inverse of ϕ_{n-1}^{-1} in (22) is computed by using the Matrix inversion lemma. See Lema 1. Let $P_n = \phi_n^{-1}$. By comparing equation (23) and (16) we find that according to the lema and following (17) we get

$$P_n = \frac{1}{\lambda}P_{n-1} - \frac{1}{\lambda}K_n u_n^T P_{n-1} \tag{25}$$

where

$$K_n = \frac{P_{n-1}u_n}{\lambda + u_n^T P_{n-1} u_n}, \tag{26}$$

is defind as the Kalman gain.

By similar manipulation we also get

$$h_n^* = h_{n-1}^* + K_n \xi_n \tag{27}$$

where

$$\xi_n = u_n^T h_{n-1}^* - d \tag{28}$$

The recursive algorithm consists on comping (25)-(28) recursively. An overview of the RLS algorithm is shown in Algorithm 1.

Algorithm 1: RLS algorithm.

Initialize: Initialize $\hat{\mathbf{\Theta}}[-1] = \mathbf{0} \in \mathbb{R}^p$, $P[-1] = \mathbb{I} \in \mathbb{R}^{p \times p}$. Set $n = 0$.

Iteration n:At the beginning of the each iteration, new observation $y[n]$ and the corresponding new row $\boldsymbol{h}[n]^T$ becomes available. We have $\hat{\mathbf{\Theta}}[n-1]$ and $\boldsymbol{P}[n-1]$. The following is computed:

1) Compute the current error
$$\bar{e}[n] = y[n] - \boldsymbol{h}[n]^T \hat{\mathbf{\Theta}}[n-1]$$
Note that $\bar{e}[n]$ is a priori error. It is calculated using $\hat{\mathbf{\Theta}}[n-1]$, not $\hat{\mathbf{\Theta}}[n]$. This is not the same with $e[n]$.
2) Compute the Kalman gain vector $\boldsymbol{K}[n] \in \mathbb{R}^{p \times 1}$
$$\boldsymbol{K}[n] = \frac{1}{\lambda^k + \boldsymbol{h}[n]^T \boldsymbol{P}[n-1]\boldsymbol{h}[n]} \boldsymbol{P}[n-1]\boldsymbol{h}[n]$$
3) Update the estimate
$$\hat{\mathbf{\Theta}}[n] = \hat{\mathbf{\Theta}}[n-1] + \boldsymbol{K}[n]\bar{e}[n]$$
4) Update $\boldsymbol{P}$:
$$\boldsymbol{P}[n] = (\mathbb{I} - \boldsymbol{K}[n]\boldsymbol{h}^T[n])\boldsymbol{P}[n-1]$$

Algorithm 2: ADAM optimizer.

Initialize: Set the initial values for the filter coefficients $\mathbf{h}[0]$. Set the initial values $\mathbf{m}_0 = \mathbf{v}_0 = 0$ for the first and second moment estimates. Initialize the step size α, the decay rates β_1, β_2 and ϵ by suggesting by the authors.

for $n = 1$ to N **do**
  $equation(35)$
  $\boldsymbol{m}_n = \beta_1 \boldsymbol{m}_{\boldsymbol{n-1}} + (1-\beta_1)\boldsymbol{g}_n$
  $\boldsymbol{v}_n = \beta_2 \boldsymbol{v}_{\boldsymbol{n-1}} + (1-\beta_2)\boldsymbol{g}_n^2$
  $\hat{\boldsymbol{v}}_n = \boldsymbol{m}_n/(1-\beta_1^n)$
  $\hat{\boldsymbol{v}}_n = \boldsymbol{v}_n/(1-\beta_2^n)$
  $\boldsymbol{h}[\boldsymbol{n}] = \boldsymbol{h}[\boldsymbol{n-1}] - \alpha\hat{\boldsymbol{m}}_n/(\sqrt{\hat{\boldsymbol{v}}_n} + \epsilon)$
end for
return $\boldsymbol{h}[N]$

B. ADAM optimizer

The ADAM optimizer uses the same basic thought as the RLS with a cost function C_2 defined as in plain LMS

$$f(\mathbf{h}[n-1]) = \frac{1}{2}e[k]^2, \tag{29}$$

where e[k] is the estimation error given by

$$e[n] = d[n] - \mathbf{h}^T[n-1]\mathbf{y}[n] \tag{30}$$

where the parameters are the same as in (7), (8), (9) and (10). Except that A is defined as y. The gradient of $f(\mathbf{h}[n-1])$ is calculated the same way that the L_2 norm was but with respect to h[n-1]

$$\begin{aligned}
|||e||_2^2 &= ||d - \mathbf{h}^T[n-1]\mathbf{y}||_2^2 && (31)\\
&= (d - \mathbf{h}^T[n-1]\mathbf{y})^T(d - \mathbf{h}^T[n-1]\mathbf{y}) && (32)\\
&= d^T d - \mathbf{y}^T\mathbf{h}[n-1]d - d^T\mathbf{h}^T[n-1]\mathbf{y} && (33)\\
&+ \mathbf{y}^T\mathbf{h}[n-1]\mathbf{h}[n-1]\mathbf{y} && (34)
\end{aligned}$$

and then the derivation gives the gradient g

$$\mathbf{g} = -\mathbf{y}(d - \mathbf{h}^T[n-1]\mathbf{y}); \tag{35}$$

which is what we need for the rest of the calculations that are shown in the overview of the algorithm 2.

The calculations made used a few parameters that were suggested by the authors in [3]. They were set to $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

C. Main algorithm

The optimization was automatized by creating a script that outputs the optimal filter taps N and M when input the number of patient. code can be found in https://github.com/dikioth/1TE651. The procedure is described in Algorithm 3.

Algorithm 3: Custom algorithm.

1) Sweep values of N and M from 5 to 200 with interval of 5.
2) Find the filter taps N and M that gives the highest performance values Q_1 and Q_2.
3) Sweep values of the neighborhood of the optimal N and M of length 5. E.g $N \pm 5$ and $N \pm 5$. [1]
4) Change the optimal N and M if a new maxima was found.

III. Results

Table I shows an example of the algorithm used for patient 1 using the ADAM optimizer. The algorithm yielded in the first sweep with steps of 5 that values $M = 10$ and $N = 50$ are optimal. After a finer sweep the algorithm gave that $M = 10$ and $N = 49$ are optimal in the case of patient 1.

The ADAM optimizer was proven to be superior in performance than the RLS algorithm. Table X shows the computing time Because of the superior performance of ADAM optimizer compared to the RLS the filter taps

The optimal filter taps and the quality values for the RLS and the ADAM optimizer methods are shown in Table III and II respectively. The reconstruction using the optimal filter taps are shown in Fig.

The reconstruction with the lowest values of Q_1 and Q_2 was the signal for patient number 2. The defaults parameters of ADAM optimizer were optimized for this signal. A detail analysis and modification of these values gave $Q_1 = 0.9292$ and $Q_2 = 0.9638$. The reconstruction of the signal with the ADAM optimizer using these modified values is shown in Fig. 11. However, the values are optimal for patient 2 only.

TABLE I
OPTIMAL FILTER TAPS N AND M FOR PATIENT 1

		N										
		45	46	47	48	49	50	51	52	53	54	55
M	5	0.9519	0.9523	0.9528	0.9532	0.9535	0.9537	0.9537	0.9533	0.9528	0.9522	0.9514
	6	0.9536	0.9540	0.9544	0.9548	0.9551	0.9553	0.9552	0.9547	0.9542	0.9534	0.9526
	7	0.9574	0.9575	0.9576	0.9578	0.9578	0.9577	0.9574	0.9569	0.9563	0.9554	0.9544
	8	0.9573	0.9574	0.9574	0.9575	0.9575	0.9573	0.9569	0.9563	0.9557	0.9548	0.9539
	9	0.9579	0.9581	0.9581	0.9583	0.9583	0.9582	0.9579	0.9573	0.9566	0.9557	0.9547
	10	0.9591	0.9593	0.9594	0.9596	0.9596	0.9595	0.9592	0.9586	0.9580	0.9571	0.9561
	11	0.9589	0.9591	0.9592	0.9594	0.9596	0.9595	0.9591	0.9585	0.9578	0.9569	0.9560
	12	0.9564	0.9566	0.9568	0.9571	0.9573	0.9573	0.9569	0.9562	0.9555	0.9548	0.9539
	13	0.9517	0.9519	0.9521	0.9523	0.9525	0.9525	0.9523	0.9517	0.9512	0.9506	0.9499
	14	0.9466	0.9467	0.9469	0.9471	0.9472	0.9473	0.9472	0.9468	0.9465	0.9460	0.9454
	15	0.9438	0.9439	0.9439	0.9439	0.9439	0.9441	0.9442	0.9438	0.9435	0.9431	0.9426

Reconstruction of signals for the other patients give slighly less Q_1 and Q_2 values than those shown in II with the default values.

TABLE II
OPTIMAL FILTER TAPS USING THE ADAM OPTIMIZER. $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ AND $\epsilon = 10^{-8}$

Patient	**M**	**N**	**Q1**	**Q2**	mean time [s]
1	10	49	0.9874	0.9934	0.7
2	6	20	0.9912	0.9956	0.6
3	110	4	0.9869	0.9932	0.7
4	16	134	0.8313	0.9456	0.8
5	34	3	0.9491	0.9771	0.7
6	127	21	0.9675	0.9838	0.8
7	30	5	0.9711	0.9852	0.7
8	5	55	0.9874	0.9934	0.7

TABLE III
OPTIMAL FILTER TAPS USING THE RLS OPTIMIZER. $\lambda = 0.99$

Patient	**M**	**N**	**Q1**	**Q2**	Mean time [s]
1	10	49	0.9568	0.9814	3.7
2	6	20	0.9904	0.9950	0.9
3	110	4	0.9893	0.9944	10.9
4	16	134	0.8922	0.9467	17.4
5	34	3	0.9612	0.9864	1.6
6	127	21	0.9452	0.9727	16.3
7	30	5	0.9678	0.9835	2.2
8	5	55	0.9908	0.9951	3.7

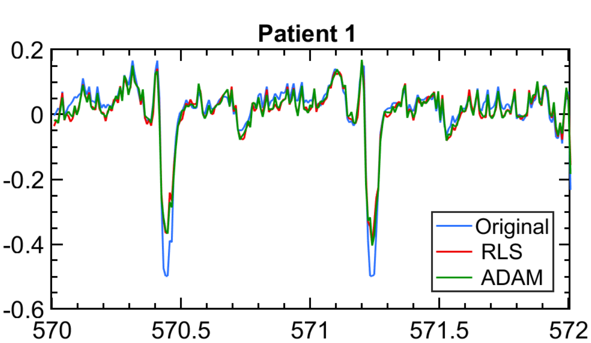

Fig. 3. Caption

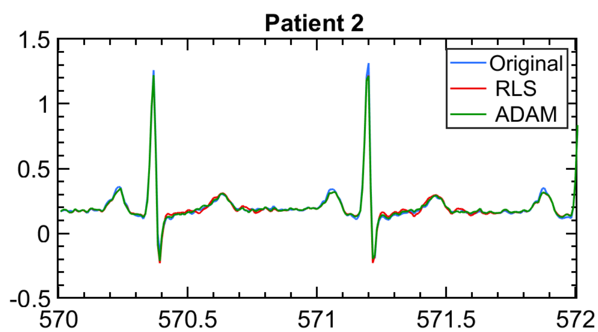

Fig. 4. Caption

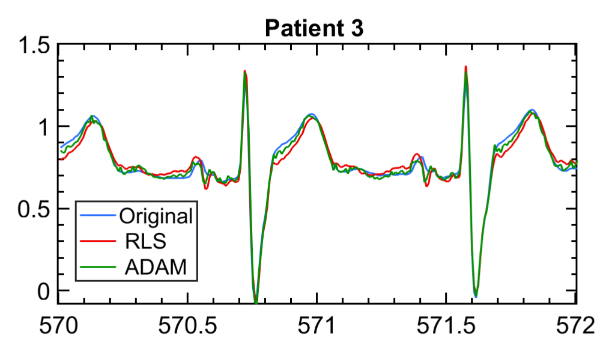

Fig. 5. Caption

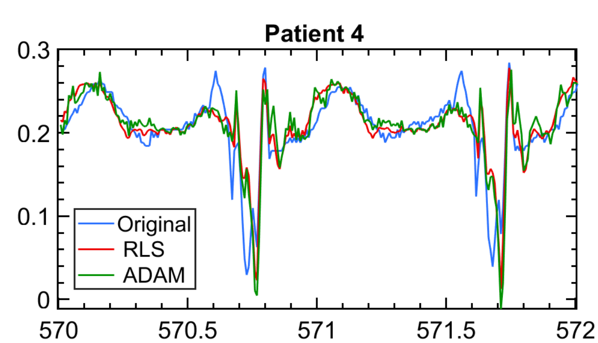

Fig. 6. Caption

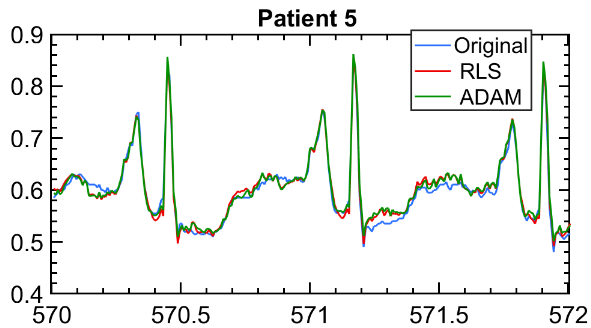

Fig. 7. Caption

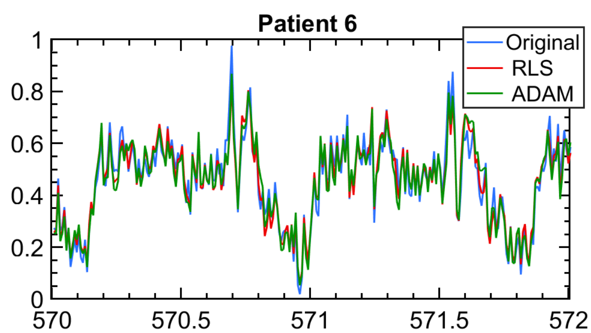

Fig. 8. Caption

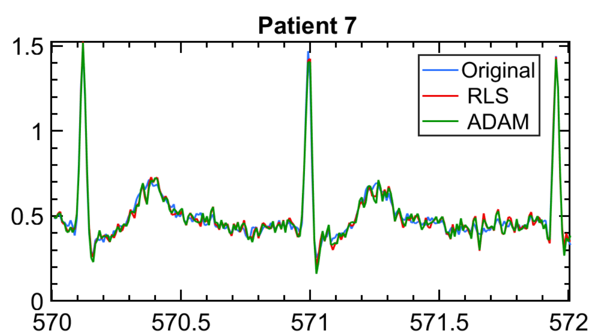

Fig. 9. Caption

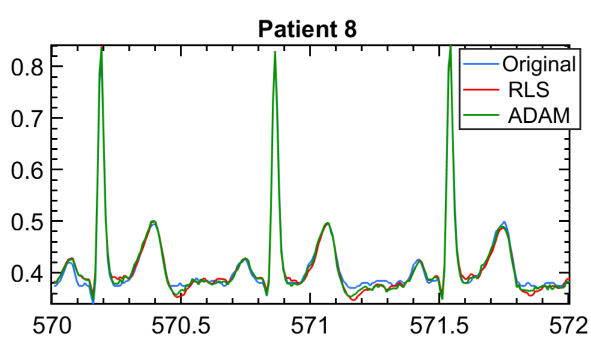

Fig. 10. Caption

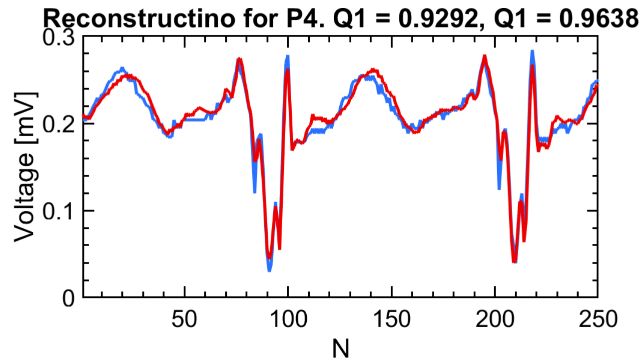

Fig. 11. Reconstructino for patient 4 using ADAM optmizer with $\alpha = 0.001$, $\beta_1 = 0.9971, \beta_2 = 0.9988, \epsilon = 1 \times 10^{-8}$

IV. Discussion

The main advantage of the ADAM optimizer over the RLS method is the time performance. It allows to test different configuration of the parameters and perform a good estimation. This method is used in various modern machine learning methods, for instance for optimization of neural networks, and has become a standard option in various software platforms.

The method used in Algorithm 3 was proven to be effective. Results gave high values of Q_1 and Q_2 and different filter taps M and M for data from different patients. The reason of the variation of N and M for different patients could be that the reference signals x_1 and x_2 are correlated different to the target signal x_T.

The finer sweep of the filter taps M and N in step 3 of the algorithm 3 was redundant for this case. The difference of Q_1 and Q_2 were small and improved the reconstruction by 0.0001% . The reason it was implemented was that no previous knowledge of Q_1 as a function of M and N was known. the assumption was that it could give strong spikes in some small intervals. Instead Q_1 and Q_2 behaved as continuous functions with respect to filter taps N and M.

The reconstruction of the missing signal for Patient 4 was the only patient that neither of the filters could replicate well with default values. When looking at the given signals x_1 and x_2, for all patients they looked less correlated to the signal x_T. When the default values for the ADAM optimizer were tuned for this specific case, an improved of 11.8% could be found. However, these values are specific for this case since no improvement could be found in the other cases. Instead, a slightly decrease on performance was seen. A stable approach to do similar reconstructions is to start with the default values suggested [3]. Then, tune the values α, β_1 and β_2. Small changes of value of ϵ does not affect the performance of the reconstruction.

References

[1] PhysioNet, "Physionet challenge: Mind the gap." [Online]. Available: https://physionet.org/content/challenge-2010/1.0.0/

[2] M. A. Woodbury, *Inverting Modified Matrices*. Princeton, NJ: Department of Statistics, Princeton University, 1950.

[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.