STEP-ADAPTIVE APPROXIMATE LEAST SQUARES

Michael Lunglmayr[†], *Christoph Unterrieder*[⋆], *Mario Huemer*[†]

[†] Johannes Kepler University Linz, Institute of Signal Processing, 4040 Linz, Austria
[⋆] Infineon Technologies AG, 9500 Villach, Austria
michael.lunglmayr@jku.at

ABSTRACT

Recently, we proposed approximate least squares (ALS), a low complexity approach to solve the linear least squares problem. In this work we present the step-adaptive linear least squares (SALS) algorithm, an extension of the ALS approach that significantly reduces its approximation error. We theoretically motivate the extension of the algorithm, and introduce a low complexity implementation scheme. Our performance simulations exhibit that SALS features a practically negligible error compared to the exact LS solution that is achieved with only a marginal complexity increase compared to ALS. This performance gain is achieved with about the same low computational complexity as the original ALS approach.

Index Terms— least squares, approximation, iterative algorithm, complexity, approximate least squares.

1. INTRODUCTION

Linear Least Squares (LS) estimation is an important approach in many areas of Signal Processing. Examples of such areas are localization [1, 2] and positioning [3], robotics [4] and sensor registration [5], power and battery applications [6, 7], biomedical applications [8, 9], or image processing [10, 11]. For the linear LS problem the following system model is assumed:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{H}$ is a known $m \times p$ observation matrix ($m \geq p$) that has full rank p and $\mathbf{y}$ is a known $m \times 1$ vector that is typically obtained from measurements. $\mathbf{x}$ is an unknown $p \times 1$ parameter vector that is to be estimated and $\mathbf{n}$ is an $m \times 1$ noise vector. The versatility of the LS approach is due to the fact that the statistical properties of $\mathbf{n}$ need not to be known. The solution $\hat{\mathbf{x}}_{LS}$ of the LS problem can be calculated as

$$\hat{\mathbf{x}}_{LS} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}. \tag{2}$$

The term $(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is often called pseudo-inverse of $\mathbf{H}$ [12]. Many signal processing applications demand solving the LS problem in realtime. Such realtime implementations are usually done using fixed point precision allowing only a limited number of arithmetic operations. Using direct solutions – often called batch solutions [13]– for $\hat{\mathbf{x}}_{LS}$ (e.g. by the pseudo inverse as shown above or QR decomposition based methods [12]) is often unfeasible in practice due to their computational complexity and their high numerical precision requirements. Many algorithms, e.g. based on the QR decomposition that are considered "low complexity" in numerical mathematics [12] are often unfeasible for realtime signal processing applications. Other algorithms, e.g. such as the one described in [14], aim at solving the normal equations that require an additional computational overhead for calculation and can lead to additional numerical problems. For realtime applications, often an approximate solution using a low number of required multiplications as well as simple hardware architectures for implementation is preferred over an exact solution obtainable only with a huge computational effort. Our aim was to develop an efficient method that can work directly on the matrix $\mathbf{H}$ without requiring the calculation of the normal equations.

As an alternative to batch solutions, iterative methods such as the iterative least squares (ILS) [12] approach can be used. This method utilizes the gradient

$$\nabla J(\hat{\mathbf{x}}) = 2\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}} - 2\mathbf{H}^T\mathbf{y} \tag{3}$$

of the squared error sum

$$J(\hat{\mathbf{x}}) = \sum_{i=1}^{m}(\mathbf{h}_i^T\hat{\mathbf{x}} - y_i)^2 = (\mathbf{H}\hat{\mathbf{x}} - \mathbf{y})^T(\mathbf{H}\hat{\mathbf{x}} - \mathbf{y}). \tag{4}$$

Here $\hat{\mathbf{x}}$ is an arbitrary estimation vector (not necessarily the best), $\mathbf{h}_i^T$ is the i^{th} row of $\mathbf{H}$ and y_i is the i^{th} element of $\mathbf{y}$, respectively. The above gradient is used in ILS for a steepest descent approach to find the minimum of $J(\hat{\mathbf{x}})$ by iteratively calculating

$$\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} - \mu\nabla J(\hat{\mathbf{x}}^{(k-1)}), \tag{5}$$

with the iteration step width μ. An optimal value of μ can be found e.g. via the singular values of $\mathbf{H}$ [12]. Rewriting (5) in the form

$$\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} - \mu\sum_{i=1}^{m}2\mathbf{h}_i(\mathbf{h}_i^T\hat{\mathbf{x}}^{(k-1)} - y_i) \tag{6}$$

indicates the required complexity of this approach. The above equation allows interpreting the gradient as a sum of partial gradients

$$d_i(\hat{\mathbf{x}}^{(k-1)}) = 2\mathbf{h}_i(\mathbf{h}_i^T\hat{\mathbf{x}}^{(k-1)} - y_i). \tag{7}$$

The calculation of a partial gradient requires $2p$ multiplications, so per ILS iteration $2pm + 1$ multiplications are required. To reduce the complexity we proposed to alternatively use a different approach that we call approximate least squares (ALS) [15]. This approach is based on an approximation of the gradient of the ILS by using only one partial gradient per iteration. As we showed in [15] and describe below, if no noise is present, ALS converges to the exact parameter vector $\mathbf{x}$. If noise is present, ALS shows a higher average error than ILS. The aim of this work is to present SALS – an advanced ALS approach – featuring a significantly lower estimation error than ALS without significantly increasing its computational complexity. We first describe the principles of ALS in the next section. We then motivate the SALS algorithm based on the theory of ALS. We then describe the SALS algorithm in detail and furthermore show a performance comparison between ALS and SALS.

2. APPROXIMATE LEAST SQUARES

Instead of calculating the sum in (6), ALS uses only a single partial gradient per iteration:

$$\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} - \mu 2\mathbf{h}_{k^\urcorner}(\mathbf{h}_{k^\urcorner}^T\hat{\mathbf{x}}^{(k-1)} - y_{k^\urcorner}). \tag{8}$$

Here the operator " $^\urcorner$" is defined such that for a positive natural number i: $i^\urcorner = ((i-1) \bmod \mathrm{m}) + 1$. For simplicity we do not write the dependence on m in the operator. In this work the divisor in the modulo operation of " $^\urcorner$" will always be the number of rows of $\mathbf{H}$. This means that if k exceeds m, the first rows of $\mathbf{H}$ and the first elements of $\mathbf{y}$ are cyclically reused again. As it is described in [15], the error of ALS can be split into two parts, one part depending on the starting error $\mathbf{e}^{(0)} = \hat{\mathbf{x}}^{(0)} - \mathbf{x}$ of the algorithm and one depending on the noise values embedded in the measurement vector $\mathbf{y}$. While the first part converges to the zero vector (when choosing μ appropriately as described below), the second part persists. As we will describe below this leads to an oscillatory behavior in the error $\mathbf{e}^{(k)} = \hat{\mathbf{x}}^{(k)} - \mathbf{x}$ for large k. To reduce the influence of this noise dependent part we introduced an averaging in the last m ALS iterations having the effect that also the error between $\hat{\mathbf{x}}^{(k)}$ and $\mathbf{x}$ is averaged over the last k iterations, as shown in [15]. This leads to the overall formulation of the algorithm as described in the following description (Algorithm: ALS), where N is the number of iterations.

Although the ALS update equation (8) is arithmetically similar to the LMS filter update equation [16], there are several significant differences. The LMS update step uses a random filter input vector and one sample of a desired signal as input, whereas the ALS update step only uses one sample y_i of the measurement vector $\mathbf{y}$ as input. Instead of the random input vectors, fixed and deterministic rows of $\mathbf{H}$ are cyclically used. A more closer connection exists to the Kaczmarz algorithm [17], as the ALS update equation (8) can be seen as a variant of the Kaczmarz algorithm for overdetermined and inconsistent linear equation systems. This cyclical usage allowed the development of the theory of ALS in a deterministic manner, instead of using stochastic arguments as in the theory of LMS. The cyclical re-use of the rows of $\mathbf{H}$ and the values of y_i also suggests the averaging in the last m operations, which in general reduces the approximation error, cf. [15].

Algorithm: ALS

```
$\hat{\mathbf{x}}_{ALS} = \mathbf{0}$
$\hat{\mathbf{x}}^{(0)} = \mathbf{0}$
**for** $k = 1 \dots N$ **do**
    $\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}^{(k-1)} + \mu 2\mathbf{h}_{k^\urcorner}(y_{k^\urcorner} - \mathbf{h}_{k^\urcorner}^T\hat{\mathbf{x}}^{(k-1)})$
    **if** $k > N - m$ **then**
        $\hat{\mathbf{x}}_{ALS} = \hat{\mathbf{x}}_{ALS} + \hat{\mathbf{x}}^{(k)}$
    **end if**
**end for**
$\hat{\mathbf{x}}_{ALS} = \frac{1}{m}\hat{\mathbf{x}}_{ALS}$
```

3. ALS ERROR ANALYSIS

With the error vector $\mathbf{e}^{(k-1)} = \hat{x}^{(k-1)} - x$ the ALS iteration can be formulated as

$$\hat{\mathbf{x}}^{(k)} = (\mathbf{I} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T)(\mathbf{e}^{(k-1)} + \mathbf{x}) + 2\mu\mathbf{h}_{k^\urcorner}y_{k^\urcorner} \tag{9}$$

$$= (\mathbf{I} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T)\mathbf{e}^{(k-1)} + \mathbf{x} + 2\mu\mathbf{h}_{k^\urcorner}n_{k^\urcorner} \tag{10}$$

with $y_{k^\urcorner} = \mathbf{h}_{k^\urcorner}^T\mathbf{x} + n_{k^\urcorner}$, and $n_{k^\urcorner}$ as the $k^{\urcorner th}$ element of $\mathbf{n}$. Subtracting $\mathbf{x}$ left and right from the equation leads to

$$\mathbf{e}^{(k)} = (\mathbf{I} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T)\mathbf{e}^{(k-1)} + 2\mu\mathbf{h}_{k^\urcorner}n_{k^\urcorner} \tag{11}$$

$$= \mathbf{M}_{k^\urcorner}\mathbf{e}^{(k-1)} + \mathbf{\Delta}_{k^\urcorner} \tag{12}$$

with $\mathbf{M}_{k^\urcorner} = (\mathbf{I} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T)$ and $\mathbf{\Delta}_{k^\urcorner} = 2\mu\mathbf{h}_{k^\urcorner}n_{k^\urcorner}$. $\mathbf{h}_{k^\urcorner}$ is an eigenvector of $\mathbf{M}_{k^\urcorner}$ to the eigenvalue $\lambda_{k^\urcorner} = (1 - 2\mu\|\mathbf{h}_{k^\urcorner}\|_2^2)$ because

$$\mathbf{M}_{k^\urcorner}\mathbf{h}_{k^\urcorner} = (\mathbf{I} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T)\mathbf{h}_{k^\urcorner}$$
$$= (1 - 2\mu\|\mathbf{h}_{k^\urcorner}\|_2^2)\mathbf{h}_{k^\urcorner} = \lambda_{k^\urcorner}\mathbf{h}_{k^\urcorner}. \tag{13}$$

We show in [15] that all other eigenvalues of $\mathbf{M}_{k^\urcorner}$ are one, respectively. That means that at iteration k a reduction of the error vector $\mathbf{e}^{(k-1)}$ – i.e. in its 2-norm – can only be achieved via the eigenvalue $\lambda_{k^\urcorner}$. This eigenvalue is affected by the step width μ as shown above. When analyzing (13), one can see that $1 > \lambda_{k^\urcorner} \geq 0$ when

$$0 < \mu \leq \frac{1}{2\,\|\mathbf{h}_{k^\urcorner}\|_2^2}. \tag{14}$$

The eigenvalue $\lambda_{k^\urcorner}$ is zero when $\mu = 1/2\ \|\mathbf{h}_{k^\urcorner}\|_2^2$.

As we also show in [15] for the noiseless case the error $\mathbf{e}^{(k)}$ converges to zero if $k \rightarrow \infty$ when

$$0 < \mu \leq \frac{1}{2 \max\limits_{i=1,\ldots,m} \|\mathbf{h}_i\|_2^2}. \tag{15}$$

Choosing the upper interval boundary

$$\mu = \frac{1}{2 \max\limits_{i=1,\ldots,m} \|\mathbf{h}_i\|_2^2} \tag{16}$$

leads to a fast decrease of the error norm [15].

4. STEP WIDTH ADJUSTMENT

Following the same arguments as above, one can see that for the first iterations the error norm reduction is even greater when using different values

$$\mu = \mu_{k^\urcorner} = \frac{1}{2\|\mathbf{h}_{k^\urcorner}^T\|_2^2}. \tag{17}$$

for every iteration $k^\urcorner$, respectively. These values $\mu_{k^\urcorner}$, with $k^\urcorner \in \{1\ldots m\}$, can be pre-calculated, adding only the memory lookup for $\mu_{k^\urcorner}$ to the complexity of the algorithm. If the rows of $\mathbf{H}$ have non-equal $2-$norms, choosing μ according to (17) leads to larger step widths then when choosing μ according to (16). Although these larger step widths result in a greater decrease of the norm of $\mathbf{e}^{(k)}$, they also result in an increase of the noise influence. By rewriting (12) as

$$\mathbf{e}^{(k)} = \mathbf{e}^{(k-1)} - 2\mu\mathbf{h}_{k^\urcorner}\mathbf{h}_{k^\urcorner}^T\mathbf{e}^{(k-1)} + 2\mu\mathbf{h}_{k^\urcorner}n_{k^\urcorner}. \tag{18}$$

we observe that an increase in μ (up to the value of (17)) has to two effects: on the one hand a greater reduction of the previous error $\mathbf{e}^{(k-1)}$ (within certain boundaries, i.e. only along the vector $\mathbf{h}_{k^\urcorner}$), one the other hand an increase of the error due to the noise value $n_{k^\urcorner}$.

Fig. 1 shows a typical error behavior of ALS when using μ according to (16). The figure shows the evolution of the error norm over the iterations k. It also allows comparing the error behavior when using $\mu_{k^\urcorner}$ according to (17) in the curve of SALS, because SALS uses this $\mu_{k^\urcorner}$ in the beginning (as it is described below). For a more detailed description of the figure we refer to the next section.

One can see in this figure that using high values of μ is beneficial for a fast decrease of the error norm in the beginning. But it also leads to a higher final error, while using a low value of μ leads to an opposite behavior. To reduce the final error, an even further reduction of μ is beneficial as is depicted by the curves of SALS. In the next section this enhanced approach is described in detail.

5. STEP-ADAPTIVE ALS

As a combination of both approaches we introduce the step-adaptive ALS (SALS). For this approach we incorporated the idea of reducing the step width with increasing k. As described before the ALS update equation has an arithmetical similarity with the update step in the LMS algorithm. Also for the LMS a reduction of the step width has been proposed, e.g. as described in [18]. However, due to the statistical derivation of the LMS, the step width reduction approaches relying on statistical measures are not feasible for the ALS algorithm. For this reason we developed a completely deterministic step width reduction approach for the ALS.

When analyzing (18) one can see that if $\mathbf{h}_{k^\urcorner}^T\mathbf{e}^{(k-1)}$ is large compared to $n_{k^\urcorner}$ higher μ values (up to the value of (17)) lead to a reduction of the error vector norm. When $\mathbf{h}_{k^\urcorner}^T\mathbf{e}^{(k-1)}$ gets smaller with increasing k, the part influenced by noise gets dominant - eventually preventing a further reduction of the error norm. If $\mathbf{h}_{k^\urcorner}^T\mathbf{e}^{(k-1)}$ and $n_{k^\urcorner}$ were known, an optimal value of μ could theoretically be calculated. But because these values are usually not known, such an approach is clearly unfeasible in practice. As an alternative we propose a practically feasible way of adapting μ for ALS.

If one analyzes the evolution of the ALS' error norm with increasing k one can observe that above a certain number of iterations the error norm shows an oscillatory behavior. This can also be seen in Fig. 1. After a certain number of iterations, the error norm of ALS shown in the figure seems to be periodic, however, exactly speaking it is still decreasing but in a practically negligible way. Although the actual error of the algorithm is unknown in practice, a detection of the algorithm's oscillation phase can easily be implemented. Due to the cyclic nature of the algorithm's access to the rows of $\mathbf{H}$ and the values of $\mathbf{y}$, the vector $\mathbf{x}^{(k)}$ (and therefore $\mathbf{e}^{(k)}$ as well as its norm) is affected by the difference $v_k = y_{k^\urcorner} - \mathbf{h}_{k^\urcorner}^T\hat{\mathbf{x}}^{(k-1)}$. For this reason, to detect an oscillation of the error norm one only has to observe this difference at every m^{th} iteration. If the oscillation phase is detected, μ is reduced at every following iteration.

Here one has to find a trade-off. If $\mu_{k^\urcorner}$ is reduced to fast the corresponding eigenvalues $\lambda_{k^\urcorner}$ are too small to reduce the previously accumulated noise error, while if μ is reduced to slow the number of iterations to obtain a desired error increases. The algorithm summarized in the following pseudocode (Algorithm: SALS) shows a low complexity and high performance approach for reducing μ. Fig. 1 also shows the evolution of the error norm of this algorithm in the SALS curve. Here one can see the reduction of the error norm for large k due to the reduction of μ. The results shown in this figure have been obtained for a 100×10 $\mathbf{H}$ matrix with random entries sampled from a $[0,1]$ uniform distribution. Also the vector $\mathbf{x}$ has been sampled from the same distribution. The simulation has been done with white Gaussian noise with a standard deviation $\sigma = 1e^{-3}$. The figure shows

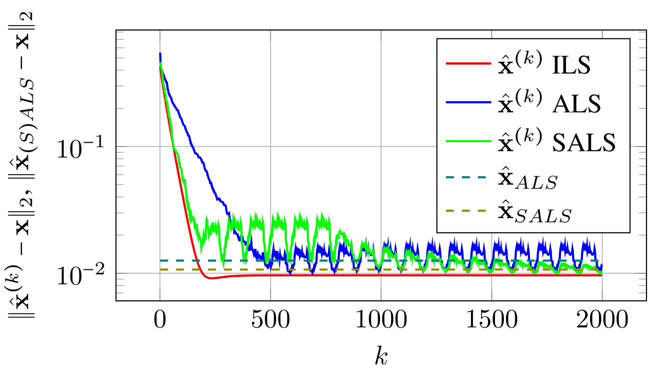

Fig. 1: Error norms over iterations k.

the error norms $\|\hat{\mathbf{x}}^{(k)}-\mathbf{x}\|_2$ at each iteration k as well as the error norm of the algorithms' output $\|\hat{\mathbf{x}}_{ALS}-\mathbf{x}\|_2$, as well as $\|\hat{\mathbf{x}}_{SALS}-\mathbf{x}\|_2$. For simplicity the error norms $\|\hat{\mathbf{x}}_{ALS}-\mathbf{x}\|_2$ and $\|\hat{\mathbf{x}}_{SALS}-\mathbf{x}\|_2$ are depicted via horizontal lines, but we want to emphasize the fact that both values are only available after the last iteration of the respective algorithms.

Algorithm: SALS

```
$\hat{\mathbf{x}}_{SALS} \leftarrow \mathbf{0}$
$\hat{\mathbf{x}}^{(0)} \leftarrow \mathbf{0}$
$v_k \leftarrow 0$
$v_{k-m} \leftarrow 1$
DontReduceMu ← True
**for** $k = 1 \dots N$ **do**
    $v_k \leftarrow y_{k^\urcorner} - \mathbf{h}_{k^\urcorner}^T \hat{\mathbf{x}}^{(k-1)}$
    **if** DontReduceMu **then**
        $\mu \leftarrow \mu_{k^\urcorner}$ according to (17)
        **if** $k^\urcorner = 1$ **then**
            **if** $|v_k - v_{k-m}| < v_{th}$ **then**
                DontReduceMu ← False
                $\mu \leftarrow \frac{1}{2 \max_{i=1\dots m} \|\mathbf{h}_i^T\|_2^2}$
            **end if**
            $v_{k-m} \leftarrow v_k$
        **end if**
    **else**
        $\mu \leftarrow f(\mu)$
    **end if**
    $\hat{\mathbf{x}}^{(k)} \leftarrow \hat{\mathbf{x}}^{(k-1)} + \mu 2 \mathbf{h}_{k^\urcorner} v_k$
    **if** $k > N - m$ **then**
        $\hat{\mathbf{x}}_{SALS} \leftarrow \hat{\mathbf{x}}_{SALS} + \hat{\mathbf{x}}^{(k)}$
    **end if**
**end for**
$\hat{\mathbf{x}}_{SALS} \leftarrow \frac{1}{m}\hat{\mathbf{x}}_{SALS}$
```

In the SALS algorithm, v_k is compared to v_{k-m}. If the absolute difference between these two values is smaller than a threshold v_{th}, μ is first set to the (typically) lower value (16). In the following iterations μ is then further reduced by the reduction function $f(\mu)$. In the next section we show simulation results for SALS using $v_{th} = 10^{-3}$ and $f(\mu) = (1-2^{-\lfloor log_2(N)\rfloor})\mu$, both found and optimized by extensive simulations. Here again N is the number of iterations of the algorithm. This reduction function has the advantage that it can be performed with only one subtraction and a shift operation (the value of $\lfloor log_2(N)\rfloor$ can be pre-calculated). It also ensures that the step-width is slowly reduced, allowing to reduce the current error while preventing the noise dependent error to become dominant.

6. SIMULATION RESULTS

In [15] we already pointed out the significantly lower computational complexity of ALS compared to iterative least squares and sequential least squares (SLS) [13]. For ALS the values of $\mu_{k^\urcorner}$ have to be calculated for (16). So the only noteworthy overhead for SALS is to store the m values of $\mu_{k^\urcorner}$ and the value $\lfloor log_2(N)\rfloor$, respectively, as well as the operations required for the calculation of $f(\mu)$ per iteration. However, this overhead might be considered negligible.

Fig. 2 shows performance results for random $\mathbf{H}$ matrices for ALS and SALS, respectively. The entries of these matrices have been sampled from a uniform distribution out of $[0,1]$. Every simulation has been done for white Gaussian noise with zero mean and standard deviation $\sigma \in S = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, respectively. Per σ value 100 random matrices $\mathbf{H}$ and 100 random vectors $\mathbf{x}$ (with random entries also sampled from a uniform distribution out of $[0,1]$) per $\mathbf{H}$ matrix have been simulated. For every σ value the averages $\overline{\|\hat{\mathbf{x}}_{ALS}-\mathbf{x}\|_2}$, $\overline{\|\hat{\mathbf{x}}_{SALS}-\mathbf{x}\|_2}$ and $\overline{\|\hat{\mathbf{x}}_{LS}-\mathbf{x}\|_2}$ over the simulated results have been calculated. Fig. 2 shows the maximum relative increase of ALS' and SALS' averaged error norms over the averaged error norms of LS, whereas the maximization has been done over the elements of S: $r_{ALS} = \max_S \left(\frac{\overline{\|\hat{\mathbf{x}}_{ALS}-\mathbf{x}\|_2}}{\overline{\|\hat{\mathbf{x}}_{LS}-\mathbf{x}\|_2}} - 1\right)$ and $r_{SALS} = \max_S \left(\frac{\overline{\|\hat{\mathbf{x}}_{SALS}-\mathbf{x}\|_2}}{\overline{\|\hat{\mathbf{x}}_{LS}-\mathbf{x}\|_2}} - 1\right)$, respectively.

For the simulations of matrices with 100 rows, the number of iterations N was set to 2000, while for the simulations of matrices with 1000 rows the number of iterations N was set to 15000. As one can see from these results, SALS shows a significant reduction in the error norm compared to ALS. From a practical point of view – especially when consider-

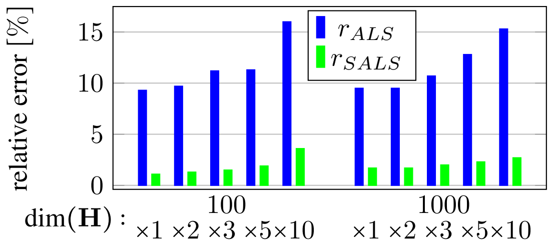

Fig. 2: Relative average errors of ALS and SALS.

ing a fixed point implementation where further errors are unavoidable – the deviation of SALS compared to LS might be considered negligible. But it is to note that for a certain group of $\mathbf{H}$ matrices, $f(\mu)$ can further be optimized which can lead to even lower approximation errors. We furthermore want to note that the relative increase of the averaged error norms remained nearly constant for each test case over all simulated σ values.

7. CONCLUSION

In this paper we introduced the step-adaptive approximate least squares algorithm, which shows a significant improvement of the recently proposed approximate least squares approach. We derived the SALS algorithm based on theoretically justified arguments of reducing the step width of ALS and presented a practically feasible step width reduction function. The performance results demonstrate significant gains compared to ALS displaying the relative error norm difference between SALS and LS in the low single digit percentage range. This performance advantage over ALS is achieved with about the same low computational complexity as the original ALS approach.

REFERENCES

[1] W. Chang, Q. Fei, S. Guangming, W. Xiaotian, "Convex Combination Based Target Localization with Noisy Angle of Arrival Measurements," *IEEE Wireless Communications Letters*, vol. 3, no. 1, pp. 14-17, February 2014.

[2] X. Yaming, Z. Jianguo, Z. Peng, "RSS-Based Source Localization When Path-Loss Model Parameters are Unknown," *IEEE Communications Letters*, vol. 18, no. 6, pp. 1055-1058, Jun. 2014.

[3] R.R. Thomas, B.T. Maharaj, B. Zayen, R. Knopp, "Multiband time-of-arrival positioning technique using an ultra-high-frequency bandwidth availability model for cognitive radio," *IET Radar, Sonar & Navigation* , vol. 7, no. 5, pp. 544-552, Jun. 2013.

[4] M. Gautier, A. Janot, P.-O. Vandanjon, "A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics," *IEEE Trans. on Control Systems Technology*, vol. 21, no. 2, pp. 428-444, Mar. 2013.

[5] S. Fortunati, A. Farina, F. Gini, A. Graziano, M.S. Greco, S. Giompapa, "Least Squares Estimation and Cramr-Rao Type Lower Bounds for Relative Sensor Registration Process," *IEEE Trans. on Signal Processing*, vol. 59, no. 3, pp. 1075-1087, Mar. 2011.

[6] C. Unterrieder, C. Zhang, M. Lunglmayr, R. Priewasser, S. Marsili, M. Huemer, "Battery state-of-charge estimation using approximate least squares," *Journal of Power Sources*, vol. 278, pp. 274-286, March 2015.

[7] C. Unterrieder , M. Lunglmayr, S. Marsili, M. Huemer, "Battery state-of-charge estimation using polynomial enhanced prediction," *IET Electronics Letters*, vol. 48, no. 21, pp. 1363-1365, Oct. 2012.

[8] J.K. Pant, S. Krishnan, "Compressive Sensing of Electrocardiogram Signals by Promoting Sparsity on the Second-Order Difference and by Using Dictionary Learning," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 8, no. 2, pp. 293-302, April 2014.

[9] L. Yuqian, D.M. Sima, S. Van Cauter, U. Himmelreich, A.C. Sava, P. Yiming, L. Yipeng, S. Van Huffel, "Unsupervised Nosologic Imaging for Glioma Diagnosis," *IEEE Trans. on Biomedical Engineering*, vol. 60, no. 6, pp. 1760-1763, Jun. 2013.

[10] S. Mihara, D. Iwai, K. Sato, "Artifact Reduction in Radiometric Compensation of Projector-Camera Systems for Steep Reflectance Variations," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1631-1638, Sept. 2014.

[11] M. Rouhani, A. Domingo Sappa, "The Richer Representation the Better Registration," *IEEE Trans. on Image Processing*, vol. 22, no. 12, pp. 5036-5049, Dec. 2013.

[12] A. Björck (1996), *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[13] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 2005.

[14] Zakharov, Y.V., Tozer, T.C., "Multiplication-free iterative algorithm for LS problem," *Electronics Letters* , vol. 40, no.9, pp. 567 - 569, April 2004.

[15] M. Lunglmayr, C. Unterrieder, M. Huemer, "Approximate Least Squares", *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4678 - 4682, Florence, Italy, May 2014.

[16] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Heam, J. R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, Dec. 1975.

[17] S. Kaczmarz, "Przyblizone rozwiazywanie ukladw rwnan liniowych. Angenäherte Auflösung von Systemen linearer Gleichungen.", Bulletin International de lAcadmie Polonaise des Sciences et des Lettres. Classe des Sciences Mathmatiques et Naturelles. Srie A, Sciences Mathmatiques, pp. 355357, 1937.

[18] T. Aboulnasr, K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulations," *IEEE Trans. on Signal Processing*, vol. 45, no.3, pp. 631-639, Mar 1997.