

Title:	Resource optimal FPGA implementation of a Least-Squares-Estimator for fast and robust online measurement of current slope and absolute value
Authors:	Andreas Liske, Simon Decker, Michael Braun
Institute:	Karlsruhe Institute of Technology (KIT) Institute of Electrical Engineering (ETI)
Type:	Conference Proceedings
Published at:	Proceedings 2019 European Conference on Power Electronics and Applications (EPE'19 ECCE Europe), Genova, Italy, 2019 Publisher: IEEE Year: 2019 ISBN: 978-9-0758-1530-6
Hyperlinks:	DOI: 10.23919/EPE.2019.8915548

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Resource optimal FPGA implementation of a Least-Squares-Estimator for fast and robust online measurement of current slope and absolute value

Andreas Liske, Simon Decker, Michael Braun
Elektrotechnisches Institut (ETI), Karlsruhe Institute of Technology (KIT)
Kaiserstr. 12, 76131 Karlsruhe, Germany
Email: Andreas.Liske@kit.edu
URL: <http://www.eti.kit.edu>

Keywords

«Estimation technique», «Measurement», «Field Programmable Gate Array (FPGA)», «Software for measurements», «Adaptive control»

Abstract

This paper presents an improved Least-Squares-Estimator for the fast, non-recursive online measurement of the first derivative and absolute value of a piecewise linear oversampled current signal. Core topic is the partially online calculation of the pseudoinverse matrix in combination with an optimized small lookup table, minimizing memory requirements significantly.

Introduction

In modern control methods for power electronics and electrical machines, the knowledge of the current slope (or derivative) of each switching state of the used inverter is a very important measurement value. It can be used for online machine parameter identification [1], modern adaptive control schemes [2, 3], condition monitoring and encoderless control [4, 5] to name just a few examples. The shape of those PWM-excited current curves in these applications can mostly be assumed to be made up of straight line segments and each switching state of the inverter produces its own current slope (see figure 1).

One possibility to get the current slope is direct measurement with specially designed current slope sensors (direct methods) [6]. Another possibility is to calculate the derivative of the absolute value of the measurement signals (indirect methods) [7, 8, 9]. Among the many mathematical possibilities to do this, the use of a Least-Squares-Estimator (LS-Estimator) has proven to be a good choice. This is mainly because the LS-Estimator is capable to not only calculate the current derivative, but at the same time filters noise and results the current's absolute value at no extra cost. A valid absolute value is absolutely necessary for any control algorithm, since every action of the control loop always is based on it. And with the fast convergence of the LS-Estimator, only few samples are sufficient for valid values, so even short measurement intervals can be used.

One very fast method for hard real-time applications is to use the pre-calculated values of the so-called pseudoinverse matrix, which are then stored in a lookup table (LUT) [10]. With this method, each sample is multiplied with its corresponding constant from the pre-calculated pseudoinverse and added to the previous result (vector product). This can be done easily and very fast in an FPGA. The downside hereby is the need for a large LUT to be implemented inside the FPGA. The memory requirement for this LUT can easily sum up to some Megabyte (MB), which is only available in large and hence expensive FPGAs or generates the need for or an external fast access memory device. A complete online calculation of the pseudoinverse would be too complex for the hard real-time demands due to the necessary matrix operations.

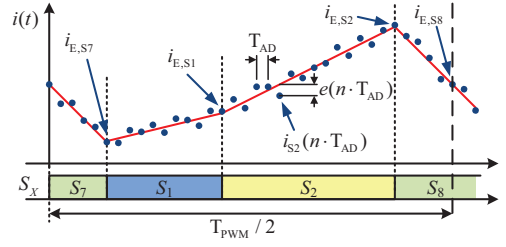


Fig. 1: PWM-excited phase current of a 3-phase synchronous machine with the ideal current waveform $i(t)$ in red and the oversampled measured values as blue dots. The switching states S_X of the first half of a PWM-period with a 3-arm modulation pattern are illustrated at the bottom.

In this paper we present an improved LS-Estimator-Algorithm that has significantly reduced memory requirements without any loss of the good real-time performance and accuracy. We mathematically show that the pseudoinverse for equidistant and oversampled measurement values consists of straight lines and hence can be calculated online in an FPGA. The former necessary large LUT that holds all the elements of the pseudoinverse can be replaced by a significantly smaller LUT which holds only the starting values and the increments of each straight line.

In the first section the LS-Estimator based on the pre-calculated pseudoinverse is explained to give a basic understanding of the operation principle. Then the elements of the pseudoinverse are investigated in more detail in the second section leading to the proposed improved implementation of the LS-Estimator. The effects of quantization errors on the new implementation method is not neglectable and is discussed in the third section. A section with measurement results of the proposed algorithm concludes the paper.

Least-Squares-Estimator based on a Lookup-Table with precalculated pseudoinverse matrixes

The current gradient and absolute value measurement with a Least-Squares-Estimator based on a Lookup-Table with pre-calculated pseudoinverse matrixes is best explained with figure 1. The current curve $i(t)$ in this figure is a typical phase current of a three-phase machine, driven by an inverter (PWM-excited current) during the first half of one PWM-period. Provided that the switching frequency is high enough, the current will rise and fall linear in straight line segments, depending on the applied switching state S_X . Although the explanation of the LS-Estimator is given in this paper with the example of a machine current, the LS-Estimator can be applied to other signals as well as long as the signal curve can be assumed to consist of straight lines. The illustration of just the first half of the PWM period is intentionally to enhance readability, but of course the algorithm can be implemented to run during each applied switching state during the whole modulation pattern. To get the current slopes of each of the applied switching states, the current sensor signal is sampled by an analog-to-digital-converter (ADC) with an equidistant time interval T_{AD} . The sample-rate $f_{AD} = 1/T_{AD}$ must be significantly higher than the PWM switching frequency f_{PWM} to make use of the LS-Estimator properly (oversampling). Hereby f_{AD} is chosen as an integer multiple of f_{PWM} . With a typical PWM switching frequency of $f_{PWM} = 8$ kHz and a sample rate of $f_{AD} = 6$ MSps, the number N_{S_X} of samples per switching state S_X depends on the applied duty cycle and varies in one half of the PWM period (see figure 1) from zero to N_{max} with

$$N_{max} = \frac{\text{max. measurement time}}{\text{sample time}} = \frac{T_{PWM}/2}{T_{AD}} = \frac{f_{AD}}{f_{PWM} \cdot 2} = \frac{6\text{MSps}}{16\text{kHz}} = 375 \quad (1)$$

Each measured value $i_{S_X}(n \cdot T_{AD})$ of the current of each switching state can be described mathematical as linear equation:

$$i_{S_X}(n \cdot T_{AD}) = i_{E,S_X} - (n \cdot T_{AD}) \cdot \left(\frac{di}{dt} \right)_{S_X} + e(n \cdot T_{AD}) \quad \text{with } n \in \{0 \dots N_{S_X} - 1\} \quad (2)$$

Note that the linear equation is set up from the end value i_{E,S_X} of each line segment. By doing this, the LS-Estimator delivers not only the current slope but also the filtered absolute value at the *end* of each switching state. This is very useful since the value at the end of a switching state is being used as setpoint value in some control schemes. The *index* n stands for the n -th *sample* in a switching state and N_{S_X} describes the *total number* of samples in each switching state. For example, a duty cycle of 60% for the switching state S_2 in figure 1 would lead to $N_{S_2} = 0.6 \cdot 375 = 225$ measured values during S_2 .

In the following the index of the switching state S_X is suppressed to improve readability. Equation (2) can be understood as an ideal model of the current as a straight line superposed with an error $e(n \cdot T_{AD})$. For each sample, one equation like (2) is given. The complete set of equations for all N samples of one switching state can be written as a set of N equations in matrix notation:

$$(i)_N = (O)_N \cdot (p)_N + (e)_N \quad (3)$$

with

$$(i)_N = \begin{pmatrix} i(0 \cdot T_{AD}) \\ i(1 \cdot T_{AD}) \\ \vdots \\ i(n \cdot T_{AD}) \\ \vdots \\ i((N-1) \cdot T_{AD}) \end{pmatrix} \quad \text{measurement vector: holds the } N \text{ measurement values} \quad (4)$$

$$(O)_N = \begin{pmatrix} 1 & -(N-1) \cdot T_{AD} \\ \vdots & \vdots \\ 1 & -n \cdot T_{AD} \\ \vdots & \vdots \\ 1 & -1 \cdot T_{AD} \\ 1 & -0 \cdot T_{AD} \end{pmatrix} \quad \text{observer matrix: depends only on } N \text{ and } T_{AD} \quad (5)$$

$$(p)_N = \begin{pmatrix} i_E \\ \left(\frac{di}{dt} \right) \end{pmatrix} \quad \text{parameter-vector: holds the searched quantities} \quad (6)$$

$$(e)_N = \begin{pmatrix} e(0 \cdot T_{AD}) \\ e(1 \cdot T_{AD}) \\ \vdots \\ e(n \cdot T_{AD}) \\ \vdots \\ e((N-1) \cdot T_{AD}) \end{pmatrix} \quad \text{error vector: holds the measurement errors} \quad (7)$$

What is of interest now is the parameter-vector $(p)_N$: Its elements are the searched absolute value i_E and the current slope (di/dt) , both filtered with the least-squares optimization criterion. As described in [10], they can be obtained by simply multiplying the so-called pseudoinverse $(O)_N^+$ of the observer matrix $(O)_N$ with the measurement vector:

$$(p)_N = \begin{pmatrix} i_E \\ \left(\frac{di}{dt} \right) \end{pmatrix} = (O)_N^+ \cdot (i)_N \quad \text{with} \quad (O)_N^+ = \left[(O)_N^T (O)_N \right]^{-1} (O)_N^T \quad (8)$$

The calculation of (8) is perfectly suited for hard real-time applications, since all that has to be done is a multiplication of all N samples of each switching state with the corresponding N elements of the pseudoinverse and adding the results. This is done sequentially and sample-based: As soon as a new measurement value is available from the ADC, the actual measurement value is multiplied with the corresponding element of the pseudoinverse and added to the result of the previous multiplication. As result, the absolute value and the current slope are available just one multiply-and-add-operation after the last measurement value of a switching state. This reduces measurement latency and hence control dead-time drastically.

To be able to do this, the necessary elements of the pseudoinverse $(O)_N^+$ must be known. An online calculation of the needed elements of $(O)_N^+$ according to (8) though is comparatively complex and generates huge computational effort due to the necessary matrix inversions. But since all elements of the observer matrix are known constants (see (5)), the pseudoinverse can be calculated prior to normal operation and stored in a lookup-table (LUT).

As already indicated by the index N , the number of elements in the pseudoinverse depends on the number of samples N in each switching state. For each individual possible length of measurement intervals for $N = 1 \dots N_{\max}$ one corresponding pseudoinverse $(O)_N^+$ has to be calculated. Since the observer matrix is a $(N, 2)$ -matrix, the pseudoinverse is a $(2, N)$ -matrix (see equations (5) and (8)). It consists of two rows, the first row contains the elements $E_{N,n}$ for the calculation of the absolute value i_E at the *end* of the considered measurement interval (hence the index 'E'), the second row contains the elements $S_{N,n}$ for the calculation of the current gradient or *slope* (index 'S'). The LUT_{PSI} (index 'PSI' stands for *pseudoinverse*) that stores all N possible pseudoinverses for all N possible lengths of measurement-vectors $(i)_N$ is hence:

$$\text{LUT}_{\text{PSI}} = \begin{bmatrix} (O)_1^+ \\ (O)_2^+ \\ \vdots \\ (O)_N^+ \\ \vdots \\ (O)_{N_{\max}}^+ \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} E_{1,1} \\ S_{1,1} \end{pmatrix} \\ \begin{pmatrix} E_{2,1} & E_{2,2} \\ S_{2,1} & S_{2,2} \end{pmatrix} \\ \vdots & \vdots & \ddots \\ \begin{pmatrix} E_{N,1} & E_{N,2} & \cdots & E_{N,n} & \cdots & E_{N,N} \\ S_{N,1} & S_{N,2} & \cdots & S_{N,n} & \cdots & S_{N,N} \end{pmatrix} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \begin{pmatrix} E_{N_{\max},1} & E_{N_{\max},2} & \cdots & E_{N,n} & \cdots & \cdots & \cdots & E_{N_{\max},N_{\max}} \\ S_{N_{\max},1} & S_{N_{\max},2} & \cdots & S_{N,n} & \cdots & \cdots & \cdots & S_{N_{\max},N_{\max}} \end{pmatrix} \end{bmatrix} \quad (9)$$

The total number M_{PSI} of entries for a LUT containing all N pseudoinverses for each possible measurement interval is

$$M_{\text{PSI}} = \frac{N_{\max}}{2} \cdot (N_{\max} + 1) \cdot 2 = N_{\max} \cdot (N_{\max} + 1) \quad (10)$$

With $N_{\max} = 375$ (see (1)) this leads to a LUT with 141000 entries (70500 for all $E_{N,n}$ and 70500 for all $S_{N,n}$). Given an exemplary word length of 32bit 4.512Mb would be necessary to store this LUT.

Least-Squares-Estimator with partially online-calculation of the pseudoinverse matrixes

To find a way to reduce the huge memory requirement for the implementation of the LS-Estimator in an FPGA, we now take a closer look onto the equations and the inner structure of the N pseudoinverses. As a first step we rewrite the observer matrix (5) with the substitution $b = -T_{\text{AD}}$ to improve the readability

of the following derivations:

$$(\mathcal{O})_N = \begin{pmatrix} 1 & (N-1) \cdot \mathbf{b} \\ \vdots & \vdots \\ 1 & 1 \cdot \mathbf{b} \\ 1 & 0 \cdot \mathbf{b} \end{pmatrix} \quad \text{with} \quad \mathbf{b} = -\mathbf{T}_{AD} \quad (11)$$

The basis for the derivation is the definition of the pseudoinverse matrix (see also (8))

$$(\mathcal{O})_N^+ = [(\mathcal{O})_N^T (\mathcal{O})_N]^{-1} (\mathcal{O})_N^T \quad (12)$$

The first thing we look at is the inner matrix product of the transposed observer matrix $(\mathcal{O})_N^T$ with the observer matrix $(\mathcal{O})_N$ that delivers matrix (A) with its elements $A_{11}, A_{12}, A_{21}, A_{22}$ as intermediate result:

$$(A) = (\mathcal{O})_N^T (\mathcal{O})_N = \begin{pmatrix} 1 & \dots & 1 & 1 & 1 \\ (N-1)\mathbf{b} & \dots & 2\mathbf{b} & \mathbf{b} & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & (N-1) \cdot \mathbf{b} \\ \vdots & \vdots \\ 1 & 1 \cdot \mathbf{b} \\ 1 & 0 \cdot \mathbf{b} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad (13)$$

$$A_{11} = 1 \cdot 1 + 1 \cdot 1 + \dots + 1 \cdot 1 = \sum_{i=1}^N 1 = N \quad (14)$$

$$A_{12} = 1 \cdot (N-1)\mathbf{b} + \dots + 1 \cdot 2\mathbf{b} + 1 \cdot \mathbf{b} + 0 = \sum_{i=1}^N (i-1) \cdot \mathbf{b} = \frac{N}{2} (N-1) \mathbf{b} \quad (15)$$

$$A_{21} = 1 \cdot (N-1)\mathbf{b} + \dots + 1 \cdot 2\mathbf{b} + 1 \cdot \mathbf{b} + 0 = \sum_{i=1}^N (i-1) \cdot \mathbf{b} = \frac{N}{2} (N-1) \mathbf{b} = A_{12} \quad (16)$$

$$\begin{aligned} A_{22} &= [(N-1)\mathbf{b}]^2 + [(N-2)\mathbf{b}]^2 + \dots + [2\mathbf{b}]^2 + [\mathbf{b}]^2 + [0]^2 \\ &= \sum_{i=1}^N [(i-1)\mathbf{b}]^2 = \sum_{i=0}^{N-1} [i \cdot \mathbf{b}]^2 = \mathbf{b}^2 \cdot \sum_{i=0}^{N-1} i^2 = \mathbf{b}^2 \cdot \sum_{i=1}^{N-1} i^2 \end{aligned} \quad (17)$$

$A_{11} = N$ as well as A_{12} and A_{21} are obviously analytically solvable. And for A_{22} , the series $\sum_{i=1}^{N-1} i^2$ is well known in mathematics and delivers as result the so called „square pyramidal number“ [11]. The n -th square pyramidal number P_n can be calculated with the formula

$$P_n = \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} \quad (18)$$

This leads to the following analytical representation of the last element A_{22} of matrix (A) (see eq.(17))

$$A_{22} = \mathbf{b}^2 \cdot \frac{N(N-1)(2N-1)}{6} \quad (19)$$

The complete matrix (A) is hence:

$$(A) = (\mathcal{O})_N^T (\mathcal{O})_N = \begin{pmatrix} N & \frac{N}{2} (N-1) \cdot \mathbf{b} \\ \frac{N}{2} (N-1) \cdot \mathbf{b} & \frac{N(N-1)(2N-1)}{6} \cdot \mathbf{b}^2 \end{pmatrix} \quad (20)$$

The next step is the inversion of matrix (A) (see eq. (12)). To prove the invertibility, the determinant is calculated:

$$\det(A) = \frac{1}{12} N^4 \mathbf{b}^2 - \frac{1}{12} N^2 \mathbf{b}^2 = \frac{1}{12} \mathbf{b}^2 (N^4 - N^2) > 0 \quad \forall N \in \mathbb{N} > 1 \quad (21)$$

This proves that matrix (A) is invertible for all measurement intervals with more than one sample, which is always given in real measurement environments. The inverse matrix $(A)_N^{-1}$ is

$$(A)_N^{-1} = \begin{pmatrix} \frac{2(2N-1)}{N(N+1)} & -\frac{6}{bN(N+1)} \\ -\frac{6}{bN(N+1)} & \frac{12}{Nb^2(N^2-1)} \end{pmatrix} \quad (22)$$

Multiplication of this inverse matrix with the transposed observer matrix now gives the searched pseudoinverse matrix (see eq. (12) and compare to eq. (9)):

$$(O)_N^+ = (A)_N^{-1} \cdot (O)_N^T = \begin{pmatrix} \frac{2(2N-1)}{N(N+1)} & -\frac{6}{bN(N+1)} \\ -\frac{6}{bN(N+1)} & \frac{12}{Nb^2(N^2-1)} \end{pmatrix} \cdot \begin{pmatrix} 1 & \dots & 1 & 1 & 1 \\ (N-1)\mathbf{b} & \dots & 2\mathbf{b} & 1\mathbf{b} & 0 \end{pmatrix} \quad (23)$$

$$= \begin{pmatrix} E_{N,1} & E_{N,2} & \dots & E_{N,n} & \dots & E_{N,N} \\ S_{N,1} & S_{N,2} & \dots & S_{N,n} & \dots & S_{N,N} \end{pmatrix} \quad (24)$$

As already mentioned above, the first row contains the elements $E_{N,n}$ for the calculation of the absolute value i_E , the second row contains the elements $S_{N,n}$ for the calculation of the current slope. Now we take a closer look unto the elements of those two row-vectors, starting with the elements $E_{N,n}$ for the calculation of the absolute value i_E :

$$E_{N,1} = \frac{2(2N-1)}{N(N+1)} \cdot 1 - \frac{6}{bN(N+1)} \cdot (N-1)\mathbf{b} = \frac{2(2N-1)}{N(N+1)} - \frac{6(N-1)}{N(N+1)}$$

⋮

$$E_{N,(n-1)} = \frac{2(2N-1)}{N(N+1)} \cdot 1 - \frac{6}{bN(N+1)} \cdot (N-(n-1))\mathbf{b} = \frac{2(2N-1)}{N(N+1)} - \frac{6(N-(n-1))}{N(N+1)} \quad (25)$$

$$E_{N,n} = \frac{2(2N-1)}{N(N+1)} \cdot 1 - \frac{6}{bN(N+1)} \cdot (N-n)\mathbf{b} = \frac{2(2N-1)}{N(N+1)} - \frac{6(N-n)}{N(N+1)} \quad (26)$$

The elements $S_{N,n}$ for the calculation of the current slope look similar:

$$S_{N,1} = -\frac{6}{bN(N+1)} \cdot 1 + \frac{12}{Nb^2(N^2-1)} \cdot (N-1)\mathbf{b}$$

⋮

$$S_{N,(n-1)} = -\frac{6}{bN(N+1)} \cdot 1 + \frac{12}{Nb^2(N^2-1)} \cdot (N-(n-1))\mathbf{b} \quad (27)$$

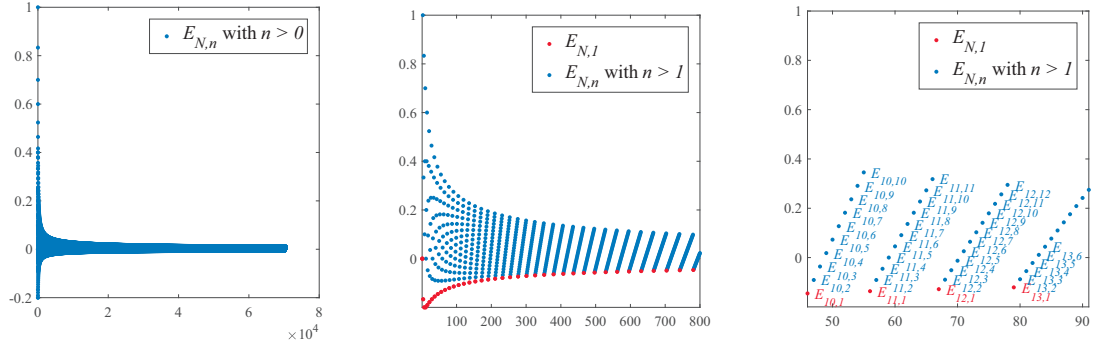
$$S_{N,n} = -\frac{6}{bN(N+1)} \cdot 1 + \frac{12}{Nb^2(N^2-1)} \cdot (N-n)\mathbf{b} \quad (28)$$

Calculating the difference between two consecutive elements in each vector analytically proves that each element of a row $E_{N,n}$ and $S_{N,n}$ in a given pseudoinverse $(O)_N^+$ has the same constant distance ΔE_N and ΔS_N to its predecessor $E_{N,n-1}$ and $S_{N,n-1}$ (see eq. (25)(26) and (27)(28)).

$$\Delta E_N(N) = E_{N,n} - E_{N,n-1} = \frac{6}{N(N+1)} \quad (29)$$

$$\Delta S_N(N) = S_{N,n} - S_{N,n-1} = \frac{12}{-b \cdot N(N^2-1)} = \frac{12}{T_{AD} \cdot N(N^2-1)} \quad (30)$$

This is illustrated in figure 2. In figure 2(a) all 70500 elements $E_{N,n}$ for the above described setup with $N_{\max} = 375$ are plotted. The elements for the $S_{N,n}$ are not plotted to enhance readability but look similar, of course with a different co-domain. The strong, exponential descent of the envelope is obvious. In 2(b) and 2(c), the x-axis was clipped to see the structure of $E_{N,n}$ in detail. Besides the mathematical proof of the validity of the above equations, this illustrates pretty well that the elements of each single pseudoin-



(a) All 70500 elements of the LUT: The strong exponential descent of the amplitudes is obvious
(b) Zoom in on the first 800 elements: the straight lines on which the elements of each vector $E_{N,n}$ are lying are becoming visible
(c) Zoom in on elements 46 to 91: those are the 4 vectors $E_{N,n}$ with $N = 10 \dots 13$ for the 4 measurement intervals with $N = 10 \dots 13$

Fig. 2: Lookup-Table LUT_{PSI} that stores all elements $E_{N,n}$ of the N pseudoinverses

verse are lying on straight lines. In figure 2(c) it is pretty good visible, that with each next pseudoinverse $(O)_{N+1}^+$, N is incremented by 1 and so the number of elements in the pseudoinverse is also 1 element more compared to its predecessor $(O)_N^+$. With an increasing N , the distance between two consecutive elements of a given pseudoinverse $(O)_N^+$ gets smaller but remains constant within $(O)_N^+$. Hence each single row vector of the pseudoinverses can be calculated not only by complex matrix operations according to equation (8), but also by a simple linear equation

$$E_{N,n} = E_{N,1} + (n-1) \cdot \Delta E_N \quad \text{with } n \in \{1 \dots N\} \quad (31)$$

$$S_{N,n} = S_{N,1} + (n-1) \cdot \Delta S_N \quad \text{with } n \in \{1 \dots N\} \quad (32)$$

All that must be known are the first elements $E_{N,1}$ and $S_{N,1}$ (marked red in figure 2(b) and 2(c)):

$$E_{N,1}(N) = \frac{4-2N}{N(N+1)} \quad (33)$$

$$S_{N,1}(N) = -\frac{6}{-b} \cdot \frac{1}{N(N+1)} \quad (34)$$

and the increments ΔE_N and ΔS_N that are given by the equations (29) and (30).

For an efficient implementation of the sample-based vector-product according to equation (8) in an FPGA, the elements of the pseudoinverses can be calculated online with the first elements $E_{N,1}$ and $S_{N,1}$ as basis and then for each following sample n by adding the corresponding increment ΔE_N and ΔS_N to its predecessor (see equations (31) and (32)).

The first elements $E_{N,1}$ and $S_{N,1}$ and the increments ΔE_N and ΔS_N can be calculated offline and stored in a now drastically smaller LUT_{SI} (index 'SI' stands for *starting value* and *increment*) in the FPGA:

$$LUT_{SI} = \begin{bmatrix} (SI)_1 \\ (SI)_2 \\ \vdots \\ (SI)_{N_{\max}} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} E_{1,1} & \Delta E_1 \\ S_{1,1} & \Delta S_1 \end{pmatrix} \\ \begin{pmatrix} E_{2,1} & \Delta E_2 \\ S_{2,1} & \Delta S_2 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} E_{N_{\max},1} & \Delta E_{N_{\max}} \\ S_{N_{\max},1} & \Delta S_{N_{\max}} \end{pmatrix} \end{bmatrix} \quad (35)$$

The total number of elements in the proposed new LUT_{SI} is

$$M_{SI} = 4N_{\max} \quad (36)$$

Advantage in terms of necessary memory space

Compared to the state-of-the-art implementation as described in section , the number of necessary elements that must be stored in a LUT in the FPGA is reduced by the factor (see eq. (10) and (36))

$$\frac{M_{PSI}}{M_{SI}} = \frac{N_{\max}(N_{\max} + 1)}{4N_{\max}} = \frac{N_{\max} + 1}{4} \quad (37)$$

In the described setup with $N_{\max}=375$ this means a massive reduction by the factor 94.

Quantization errors

Since the LS-Algorithm is calculated in an FPGA, the elements of the LUTs are stored as fixed-point numbers and quantization errors occur similarly in LUT_{PSI} as well as in LUT_{SI}. In LUT_{PSI} the maximum quantization error is simply the smallest value that is representable by the chosen fixed-point format, since every single element in LUT_{PSI} is converted to fixed-point format separately. In a fixed-point format with f fractional bits this is

$$e_{Q,PSI} = e_{Q,f} = 2^{-f} \quad (38)$$

In the proposed new algorithm this quantization error is the same for the stored elements in LUT_{SI}. But those are just the first elements and the increments, not yet the constants for the LS-Estimator. As described above, they are calculated online by adding the increments ΔE_N and ΔS_N to the calculated value of the previous sample (see eq. (31) and (32)). Hence, the quantization error of the increments ΔE_N and ΔS_N is also added repeatedly with every sample, leading to an increasing inherited error in the values of the pseudoinverse. The resulting quantization error for the LS-Algorithm with the proposed online calculation of the $(E_{N,n})$ and $(S_{N,n})$ is therefore

$$e_{Q,SI} = (N - 1) \cdot e_{Q,f} \quad (39)$$

The longer the switching state, the bigger is N and the resulting quantization error. To achieve the same accuracy in the LS-Algorithm as with the LUT_{PSI}, the precision of the ΔE_N and ΔS_N has to be improved by more fractional bits f_{SI} for these fixed-point values:

$$f_{SI} = f + \lceil \log_2(N - 1) \rceil \quad (40)$$

So with $N_{\max}=375$, up to $\lceil \log_2(374) \rceil = 9$ additional fractional bits are necessary. But even if for all elements in LUT_{SI} the additional fraction bits are implemented according to (40), the necessary memory is still reduced by a significant factor of 73, compared to an implementation with the complete LUT_{PSI} with a realistic word length of its elements of $w_{PSI}=32\text{bit}$.

Measurement results

We implemented the proposed algorithm in a Cyclone IV-FPGA from Altera (EP4CE40F23C6) which is used as modulator for the inverter in one of our institute's automotive drive test benches. The algorithm was tested in this test bench with an automotive 3-phase permanent magnet synchronous machine as application. To measure the current, an A/D-Converter (THS1206) with a resolution of 12 bit was operated at 6MSps. The switching frequency of the inverter was 8kHz. A complete description of the implementation is not the main focus of this paper and is hence not explicated further.

The values of the absolute values and the slopes are reset to zero at the beginning of each half PWM-period. The calculation of each switching state's value is then started as soon as the switching state is

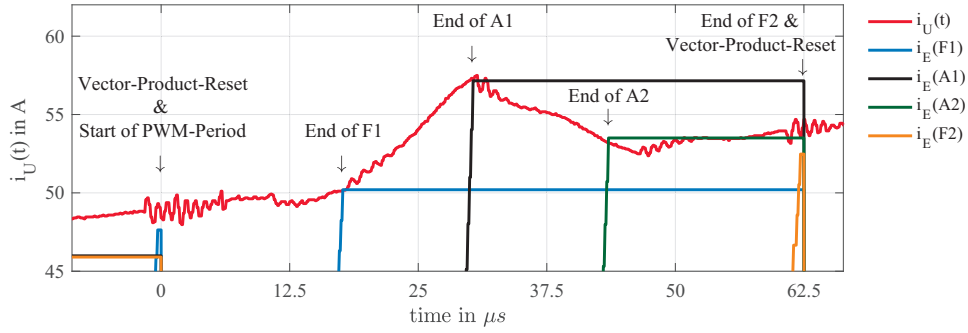


Fig. 3: Measurement results of the identification of the absolute values of the current

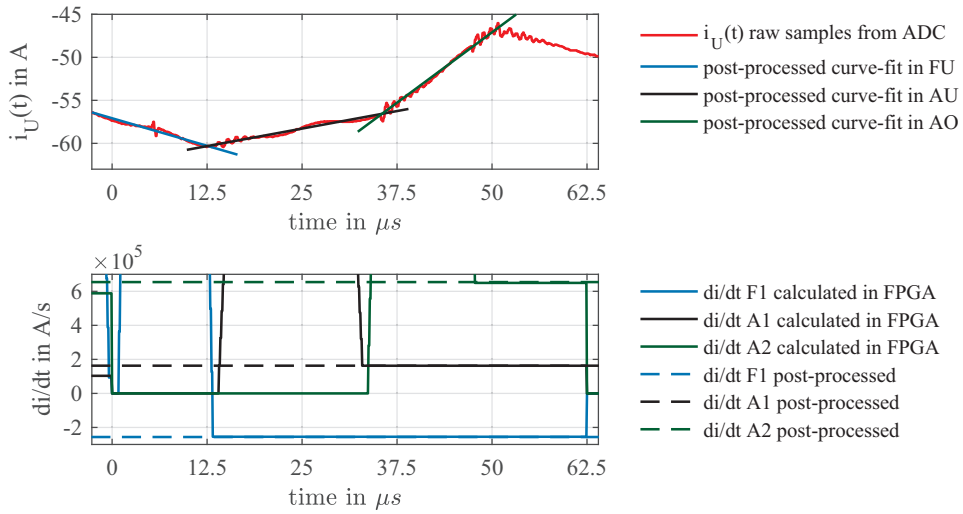


Fig. 4: Measurement results of the identification of the current slopes

active. During the switching state, the vector products for absolute value and slope as described above are being calculated with each new sample. As soon as the specific switching state ends, the values are held constant and are stored in buffers for later evaluation.

In fig. 3 the phase current $i_U(t)$ and the calculated absolute values are shown. The red curve $i_U(t)$ consists hereby only of the raw, unfiltered samples from the ADC. The measurement shows, that the calculation of the absolute values at the end of each switching period is working fine and correctly hits the value of the current curve $i_U(t)$ at the corresponding point in time.

In fig. 4, the current samples that are measured by the AD-Converter are plotted red. The blue straight line segments are post-processed linear curve-fits with Matlab's 'polyfit'-function to generate a valid value of the slopes to prove the results of the proposed algorithm that is running inside of the FPGA. Those post-processed slopes for the three switching states F1, A1 and A2 are plotted as dashed lines in the bottom picture of 4. The current slopes that are calculated by the proposed algorithm inside of the FPGA are plotted as solid lines in the corresponding color. As with the absolute values, the measurement results prove that the proposed algorithm delivers the correct values for the current slopes as well.

Conclusion

In this paper we presented an algorithm which allows an efficient and resource optimal implementation of a Least-Squares-Estimator for the fast online measurement of the slope and the absolute value of a current signal. One well-proven non-recursive LS-algorithm for fast online measurement was described, which is based on a lookup-table that contains the complete set of pseudoinverse matrixes for all possible lengths of measurement intervals. A closer look onto this lookup-table led to the new proposed algorithm. Its core topic, the partially online calculation of the pseudoinverse matrixes in combination

with an optimized, significantly smaller Lookup-Table was derived and presented. Since the algorithm is designed for FPGA-implementation, the issue of quantization errors was addressed. Measurement results from a realistic implementation inside an automotive drive test bench were presented, that prove the theory completely.

With our proposed algorithm, the memory requirement for the implementation of a non-recursive LS-Estimator is reduced significantly without any loss in accuracy and performance. The absolute value and the current slope are available just one multiply-and-add-operation after the last measurement value was available. The massive saving of valuable hardware-resources is an important step towards cost-efficient implementation of fast online slope and absolute value identification of measurement signals. This enables the implementation of advanced motor control and power electronics control schemes in common control hardware.

References

- [1] S. Decker, J. Richter, and M. Braun, "Predictive current control and online parameter identification of interior permanent magnet synchronous machines," in *2016 18th European Conference on Power Electronics and Applications (EPE'16 ECCE Europe)*, Sep. 2016, pp. 1–10.
- [2] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 12, pp. 4312–4324, 2008.
- [3] A. Liske and M. Braun, "Direct adaptive current control - a universal current control scheme for electrical machines," in *The 40th Annual Conference of the IEEE Industrial Electronics Society (IECON), 2014*, 2014, pp. 1–7.
- [4] M. Schrödl, "Sensorless control of permanent magnet synchronous motors," *Electric Machines & Power Systems*, vol. 22, no. 2, pp. 173–185, 1994.
- [5] P. Landsmann, D. Paulus, P. Stolze, and R. Kennel, "Saliency based encoderless predictive torque control without signal injection," in *Power Electronics Conference (IPEC), 2010 International*, 2010, pp. 3029–3034.
- [6] S. Decker, A. Liske, D. Schweiker, J. Kolb, and M. Braun, "Measurement of two-level inverter induced current slopes at high switching frequencies for control and identification algorithms of electrical machines," in *2018 International Power Electronics Conference (IPEC-Niigata 2018 - ECCE Asia)*, May 2018, pp. 2848–2853.
- [7] Y. Duan and M. Sumner, "A novel current derivative measurement using recursive least square algorithms for sensorless control of permanent magnet synchronous machine: 2 - 5 june 2012, harbin, china ; [also referred to as] ecce asia [as a part of ecce (ieee energy conversion congress and exposition) worldwide series]," pp. 1193–1200, 2012.
- [8] A. Liske, F. Stamer, and M. Braun, "Easy current slope detection for low cost implementation of the direct adaptive current control for dc-dc-converters," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. Piscataway, NJ: IEEE, 2015, pp. 180–186.
- [9] J. Wang and A. Binder, "Current slope calculation in fpga for sensorless control technique and associated slope based predictive precise current control," in *2013 IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)*. Piscataway, NJ: IEEE, 2013, pp. 1–8.
- [10] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, ser. Classics in applied mathematics. Philadelphia, Pa: Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 1995, vol. 15.
- [11] J. H. Conway and R. K. Guy, *The book of numbers*. New York, NY: Copernicus, 2006.