Implementation of a Novel Architecture for DFT-based Channel Estimators in OFDM Systems

Stala, Michal; Edfors, Ove; Öwall, Viktor

Published in:
[Host publication title missing]

2014

Implementation of a Novel Architecture for DFT-based Channel Estimators in OFDM Systems

Michal Stala, Ove Edfors, Viktor Öwall
Lund University
Email: Michal.Stala, Ove.Edfors, Viktor.Owall@eit.lth.se

***Abstract*—A new architecture for Discrete Fourier Transform (DFT) based channel estimation has been analyzed, implemented and synthesized for ASIC. The core concept of the proposed estimation algorithm is to process the channel increments rather than the channel coefficients. With strong enough time correlation, we can reduce the wordlength of processing blocks compared to standard channel estimators and hence the resulting area and power. We provide an analytical tool to predict the potential gains in bit reduction for different mobility scenarios. Our simulations show that the wordlength can be reduced from 9 to 3 bits when operating in low mobility scenarios, with 5Hz Doppler frequency, while maintaining acceptable performance. Synthesis results show up to 40% reduction in area, compared to the original DFT-based approach, in a 65nm CMOS process.**

I. Introduction

Many recent high-speed wireless communication techniques are based on OFDM, such as Wireless Local Area Network (WLAN), 3GPP Long-term Evolution (LTE), Digital Audio Broadcasting (DAB) and Digital Video Broadcasting (DVB) [1] [2]. A basic OFDM system is depicted in Fig. 1, which includes the transmitter (upper part), channel, and receiver (lower part). An important block in the receiver chain is the channel estimator, where the signal path (channel) from the transmitter to the receiver is estimated. Analysis [3] shows that 35-50% of the total base-band processing complexity (including memories) in a typical LTE receiver can be attributed to channel estimation and equalization. Therefore, it is of importance to find as efficient channel estimation implementations as possible.

Channel estimation in OFDM systems is a fairly old topic, with many different algorithms proposed over the years, e.g. [4] [5]. Channel estimation is most often based on known data on certain subcarriers at certain time instants, so called pilot signals. Channel estimation can be done directly in the frequency domain, by filtering and interpolating between the pilot positions. Another popular method is to convert the estimated channel values at pilot positions to the time domain, perform a windowing, and convert the estimates back to the frequency domain again. This method is often referred to as *DFT based channel estimation* [4] and is motivated by the possible complexity reductions. After channel estimation, the received signal is equalized (EQ) in the frequency domain and de-mapped.

The channel estimation method proposed in this paper, which is based on the work in [6], follows the principle of the DFT-based channel estimators, but further reduces complexity in slow-fading channels by exploiting the strong correlation between consecutive channel estimates. By processing the channel increment, rather than the channel itself, we obtain a signal with lower dynamic range that can be processed with a shorter wordlength without significant loss of performance. A property of the proposed method is that it contains feedback paths, which means that quantization errors will accumulate in a fixed point implementation. Regular channel estimation, for example a normal DFT based estimation, is therefore required occasionally to reset the accumulated errors. There are still several practical scenarios where this proposed algorithm can be an efficient alternative to traditional channel estimation.

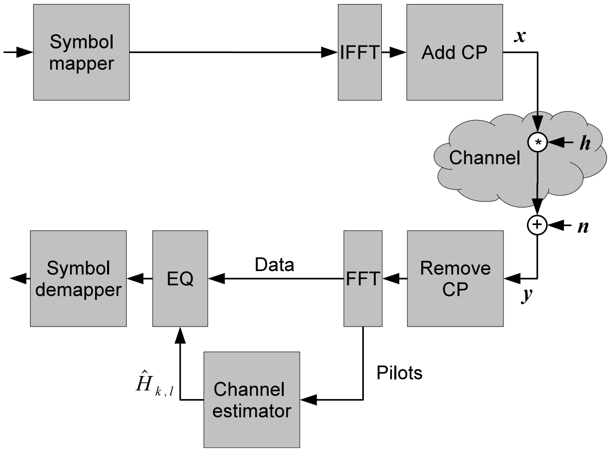

Fig. 1. **OFDM system overview**

The approach of processing the increment is most efficient in slow-fading scenarios, while mobile communication is often performed in a mix of scenarios. If the algorithm can be implemented with small enough foot-print, in terms of silicon area, it can serve as an extra, low power hardware accelerator for the low Doppler scenarios. With such an accelerator, a wireless terminal can reduce the power consumption of its base-band processing when used at home, the office, or other low mobility environments.

In more detail, a terminal using the proposed algorithm should be implemented with the possibility for a full channel estimation that could be multiplexed occasionally to avoid error propagation. The hardware for the full channel estimator could either be dedicated hardware, a software solution (Digital Signal Processor) or other hardware such as the main

FFT intended for the normal transformation to the frequency domain, see Fig. 1. A small accelerator, like the one proposed, can therefore save power without increasing the total area significantly.

As mentioned above, the concepts described in this paper are based on the DFT class of channel estimators and two variants, a standard approach and a new solution, are compared against each other. The estimators are described in more detail in Section III and both hardware implementation and synthesis results are described in Section IV. Conclusions and acknowledgements are finally presented in sections VI and VII, respectively.

II. System model

A system model, along the lines of [2], including the processing blocks as well as the channel, is shown in Fig. 1. In the OFDM transmitter, the symbol mapper converts the input bitstream to symbols. The IFFT converts the symbols to time domain and a Cyclic Prefix (CP) is added to avoid Inter Carrier and Inter Symbol Interference (ICI & ISI). The resulting signal is transmitted through the Channel and the received data is expressed as

$$y[i] = (h * x)[i] + n[i], \tag{1}$$

where h is the the channel impulse response, x is the transmited signal, n is the noise, i is the time index, and '$*$' denotes convolution. Throughout the paper bold lowercase, bold uppercase, and normal letters indicate vectors, matrices, and scalars, respectively. The receiver processes the received signal by first removing the CP and then converting the data to the frequency domain using the FFT. The channel estimator typically estimates $\boldsymbol{h}$ using frequency domain data. The resulting channel estimate $\hat{H}_{k,l}$ (in frequency domain) is passed to the Equalizer (EQ), here k is the OFDM symbol time index and l the subcarrier index. The final step is to demap the data from a given constellation to a bit stream. This model gives an overview of the OFDM system while the following sections will focus on one block in the model – the channel estimator.

The research presented in this paper is focused on the 3GPP LTE standard and in the evaluations we use the 5 Hz Doppler EPA and 70 Hz Doppler EVA channel models as specified in [7].

III. Time difference based channel estimator

This section will introduce the concept of channel estimation in an OFDM system followed by a description of a DFT based channel estimation. Finally, the time difference based channel estimation is explained.

A. Channel estimation

Channel estimation in an OFDM system is typically performed after the FFT, based on frequency domain pilot data, as shown in Fig. 1. The receiver has information of what the transmitter has sent (pilot data) and is able to estimate the channel ($\hat{H}_{k,l}$) by comparing the received data $Y_{k,l}$ with the known sent data $X_{k,l}$ based on the following frequency domain model of the system

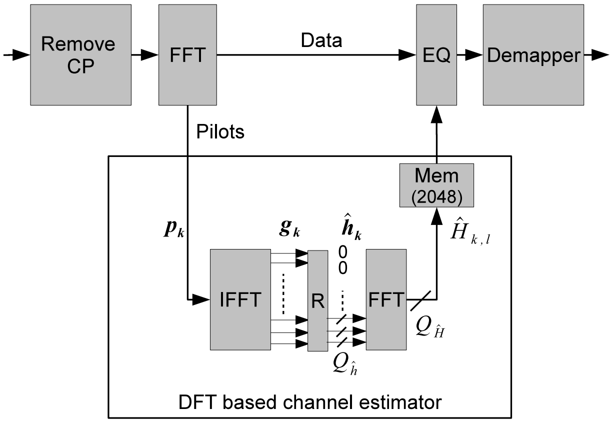

Fig. 2. **OFDM receiver including a DFT based channel estimator**

$$Y_{k,l} = H_{k,l} X_{k,l} + N_{k,l}, \tag{2}$$

where k and l are defined as before (time and subcarrier indices) and $N_{k,l}$ denotes the noise contribution in the frequency domain.

B. DFT based channel estimation

In a DFT based channel estimator, conversion of the pilot data, from frequency back to time is performed by the IFFT. There are several options for the processing in time domain [4], marked as 'R' in Fig. 2 which also can be seen as a linear transformation. The simplest approach is to use a time window where parts of the data (the output of the IFFT) if forced to zero according to

$$\hat{\boldsymbol{h}}_{\boldsymbol{k}} = \boldsymbol{R}\boldsymbol{g}_{\boldsymbol{k}}, \tag{3}$$

where

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{I}_{\boldsymbol{M}\times\boldsymbol{M}} & 0 & ... & 0 \\ 0 & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 \end{bmatrix} \tag{4}$$

is a limited identity matrix. This time windowing can also be interpreted as a low-pass filtering in the frequency domain. The length of the impulse response is in this paper set to the length of the normal cyclic prefix in LTE ($M = 144$) [8]. This simple windowing/filtering approach is assumed below. Other approaches could be used but the analysis of the wordlength reduction would not be affected to a large extent. The maximum number of sub-carriers is 2048 and the number of pilots per stream in LTE is 200. These numbers are presented to give a relation to the length of the cyclic prefix. The FFT sizes are also dependent on these numbers. The FFT after the CP removal block is 2048 points as well as the FFT in the channel estimator. The IFFT in the estimator could be implemented as a 256 point IFFT since there are only 200 pilots.

The memory that holds the $\hat{H}$ values is needed in both the original DFT method and the proposed method and is used for time interpolation of the channel estimates. Further, the pilot data in frequency domain is defined as

$$\boldsymbol{g_k} = \text{IFFT}(\boldsymbol{p_k}), \tag{5}$$

where $\boldsymbol{p_k}$ is the pilot symbols in frequency domain and k is the OFDM symbol number. The two quantization points, $Q_{\hat{h}}$ and $Q_{\hat{H}}$, that are shown in Fig. 2, in addition to the different signals, are used and described in Sections III-E and IV.

C. Proposed method

In [6] we have introduced the theory for an alternative method of performing the DFT based channel estimation which can theoretically improve the area and power consumption in a hardware implementation. In this paper we further investigate, through an actual ASIC implementation, that improvements can be observed in a real implementation and we also introduce an analytical tool to predict the wordlength reduction in the FFT of the channel estimator.

The proposed method makes use of the high time correlation of channel estimates. Instead of processing the pilot data directly, as in the original algorithm, the difference between the two occurrences (in time) of pilot data, according to Fig. 3, is processed by the FFT according to

$$\boldsymbol{\Delta\hat{h}_k} = \boldsymbol{\hat{h}_k} - \boldsymbol{\hat{h}_{k-\Delta k}}, \tag{6}$$

where $\boldsymbol{\hat{h}_k}$ is defined in Eq. 3 and Δk is the distance in OFDM symbols between the pilot positions in time. In LTE, Δk can either be 3 or 4 depending on the OFDM symbol number [8].

To provide a function for the theoretical bit reduction potential some basic concepts and equations need to be presented. For a Jakes-fading channel, the correlation between two samples of the channel, h_k and $h_{k-\Delta k}$, spaced T seconds apart, at a maximal Doppler shift of f_d Hz is

$$\rho(T) = \sigma_h^2 J_0(2\pi f_d T), \tag{7}$$

where σ_h^2 is the variance of the channel samples and $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind [9]. If we were to process the channel samples directly, we would have to deal with the full variance σ_h^2. By processing the difference signal

$$\boldsymbol{\Delta h_k} = \boldsymbol{h_k} - \boldsymbol{h_{k-\Delta k}}, \tag{8}$$

instead, we only have to deal with a variance

$$\mathbb{E}|\ \Delta h\ |^2 = \sigma_h^2 2(1 - J_0(2\pi f_d T)). \tag{9}$$

The ratio between the variances of the difference and the original channel samples become

$$\frac{\sigma_h^2 2(1 - J_0(2\pi f_d T))}{\sigma_h^2} = 2(1 - J_0(2\pi f_d T)). \tag{10}$$

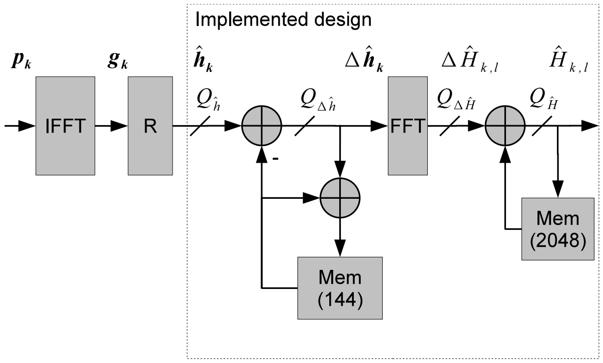

Fig. 3. **Proposed channel estimator**

For each 6.02 dB reduction of the variance, one bit less is required to maintain a fixed quantization-noise level. Therefore we estimate the potential difference in worldlength, when changing to processing of the differential signal, to be around

$$v(f_d T) = \frac{10\log_{10} 2(1 - J_0(2\pi f_d T))}{6.02}, \tag{11}$$

which we show in Fig. 4 for a range relative Dopplers. The relative Doppler $f_d T$ in (11) is defined through standard LTE parameters as

$$f_d T = f_d T_s \Delta k, \tag{12}$$

where $T_s = 70\mu s$ is roughly the LTE symbol time (varies somewhat) and $\Delta k = 4$ is the longest distance between pilots, measured in LTE symbols. The *no-gain region* in Fig. 4 shows where we do not expect any gains by processing the differential signal. Two Doppler frequencies of special interest, 5Hz and 70Hz, are indicated in the figure since these are the ones used in the performance simulations in Section IV. However, it should be noted that the analysis is somewhat rough. For instance, it does not take into account the effect of taking the difference between samples with noise, which is what we do in a real situation. Realistic reductions in wordlength should therefore be somewhat smaller than prediced in Fig. 4, which is also shown in Sec. IV.

The analysis shows that the wordlength of the FFT has the potential to be significantly reduced in low mobility scenarios, if the difference is processed rather than the channel samples directly. Some additional computations, compared to the original DFT solution, must be introduced and other considerations such as wordlength selection and noise contributions must be taken into account, which is analyzed throughout this paper. If we include noise contributions in (6) we get

$$\boldsymbol{\Delta\hat{h}_k} = \boldsymbol{\hat{h}_k} + \boldsymbol{n_k} - (\boldsymbol{\hat{h}_{k-\Delta k}} + \boldsymbol{n_{k-\Delta k}}), \tag{13}$$

and it can be noted that the variance of the noise in $\boldsymbol{\Delta\hat{h}_k}$ will become twice as large, assuming the noise terms, $\boldsymbol{n_k}$ and $\boldsymbol{n_{k-\Delta k}}$ are uncorrelated. The proposed method must therefore be able to handle this extra noise contribution without significant performance degradation.

Since the difference $\boldsymbol{\Delta\hat{h}_k}$ is processed by the FFT, instead of $\boldsymbol{\hat{h}_k}$, the wordlength of the FFT can be reduced, yielding area and power reductions. The fact that lower Doppler gives higher correlation makes this method more suitable in low Doppler

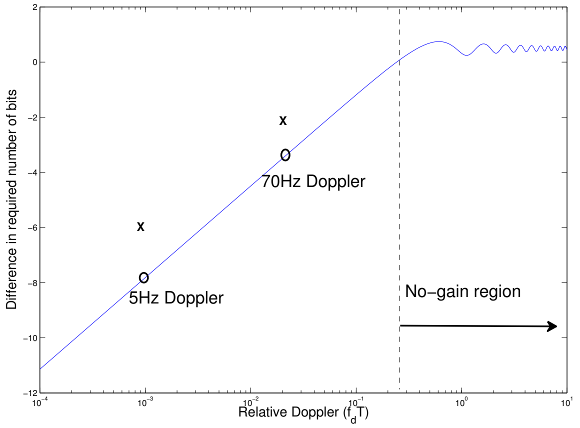

Fig. 4. **Potential bit reductions for different relative Doppler frequencies (X's marks the actual bit reductions described in Section IV)**

scenarios as shown in Fig. 4. The performance analysis in Section IV is threfore mainly focused on an EPA 5Hz Doppler channel as defined in the 3GPP specification [7].

The IFFT and the 'R' blocks are left unchanged compared to the original DFT approach as seen in Fig. 2 and Fig. 3. The FFT is still present but the input, $\boldsymbol{\Delta\hat{h}}$, has a reduced number of bits compared to $\boldsymbol{\hat{h}}$ that is processed in the original DFT solution shown in Fig. 2. The additional hardware blocks required in the new approach are the 3 adders and the memory that stores the channel estimates for the previous time instance. The size of this memory should be the length of the impulse response of the channel, in this case 144 samples.

The feedback could be extended to also cover the IFFT block in Fig. 3 reducing the hardware complexity even further. The focus of this work has been of reducing the size of the FFT, which consumes the major part of the channel estimator, i.e. 2048 points compared to the IFFT which is 256 points.

D. Extension of the proposed method

It might be tempting to perform another discrete derivation, to reduce the word length even more, resulting in

$$\boldsymbol{\Delta^2\hat{h}_k} = \boldsymbol{\Delta\hat{h}_k} - \boldsymbol{\Delta\hat{h}_{k-\Delta k}}. \tag{14}$$

However, the noise contribution in $\boldsymbol{\Delta^2\hat{h}_k}$ increases considerably, and it can be shown that the variance is six times larger than the original solution which makes this solution unpractical to implement.

E. Quantization

A fixed point hardware implementation requires a wordlength analysis for different parts of the algorithm. Both figures 2, and 3 show quantization points for the original and the new approach. The previous investigation in [6] showed that in the case of LTE, the input to the FFT could be reduced from 7 bits, for the original method, to 5 bits in the proposed method. This paper describes a complete fixed point implementation of the algorithm and shows slightly different quantization levels compared to the theoretical results in [6]. Quantization levels of interest are: the internal quantization of data in the FFT and the input and output of the FFT. These quantization levels together with architectural options are discussed in the next section.

IV. Performance analysis

Throughout this work, standard approaches for the implementation of the FFT are used with the main purpose to analyze if the gain due to wordlength reduction remains in a full fixed-point hardware implementation [10]. A comparison between the standard and the proposed DFT based channel estimation is performed, with wordlengths that give limited performance degradation in terms of BER, in the different stages in the algorithm. The hardware architectures in this paper have been developed in synthesizable C++ and synthesized to VHDL with CatapultC and to netlist with Synopsis Design Compiler in a standard 65nm process.

The FFT architecture is based on a radix-2 pipelined architecture [10] [11], in which the internal quantization of the FFT is static with increasing wordlength in each stage [12]. All simulations in this paper have been performed with standard Rayleigh fading channels (EPA and EVA) specified for LTE by 3GPP [7] and with 64QAM constellation. Simulations for the original DFT method (DFT_{std}) presented in Fig. 5, show that the input ($Q_{\hat{h}}$) and output ($Q_{\hat{H}_{DFT}}$) quantization levels before and after the FFT should both be 9 bits (marked with 9/9 in legend) with reasonable performance degradation compared to floating point (solid line), in an EPA 5Hz Doppler scenario. Perfect channel state information (CSI) simulations are also shown (dashed line) to give a baseline for the channel estimator. The proposed method, marked with DFT_{Δ}, has a quantization of 3 bits for the input ($Q_{\Delta\hat{h}}$) and 4 bits for the output ($Q_{\Delta\hat{H}}$), marked as 3/4 in the Fig. 5. The reduction of bits from 9 to 3/4 is consistent with the theoretical results shown in Fig. 4. The difference between the theoretical and the actual simulations can be explained with the fact that the theoretical model does not include noise and uses perfect channel knowledge rather than estimates of the channel. The process of determining required wordlengths with acceptable performance degradation is performed in a step-wise process. Initially only the input and output are quantized while floating point precision is kept internally in the FFT, those results are shown in Fig. 5. Later, a fully fixed-point implementation show additional performance degradation. However, the internal wordlengths can be chosen to limit the additional degradation. Fig. 7 illustrates this extra degradation where the partially fixed point simulations from Fig. 5 (but with fewer simulation points) are compared with fully fixed point simulations. Scenarios with higher Doppler frequency will naturally show less difference in wordlengths between the two methods, as seen in Fig. 4.

Fig. 6, where a 70Hz Doppler channel is simulated, shows that a quantization to 7 bits in both input and output is required to obtain the same performance as the standard DFT method with 9 bit quantization levels. This result also corresponds well

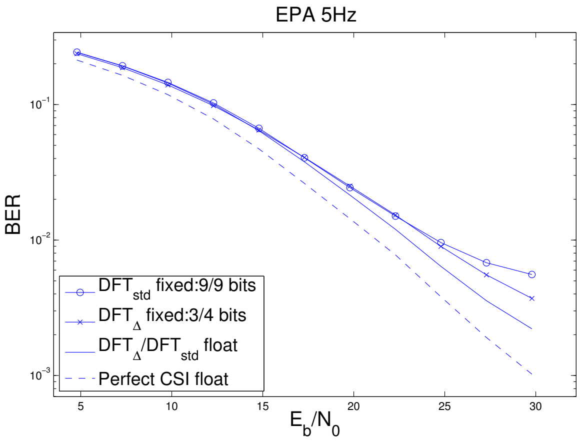

Fig. 5. **Performance simulations: 64QAM 5Hz Doppler frequency**

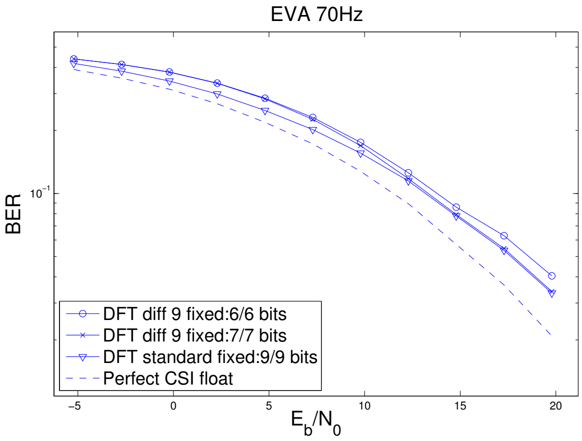

Fig. 6. **Performance simulations: 64QAM 70Hz Doppler frequency**

with the theoretical results shown in Fig. 4. The gain in area, in the 70 Hz Doppler case, is too small to motivate a separate hardware accelerator.

Fig. 7 shows results from a complete fixed point simulation and a partially fixed point simulation where only input and output are quantized. The details of the internal quantization levels are described in detail in Section V.

A. Limitations

A requirement for the proposed fixed-point implementation is that a complete DFT based channel estimation is done (without processing the difference) in certain intervals. This is to limit the effects of divergence and instability caused by the accumulation of quantization errors in the recursive structures of the proposed method. All simulations in this paper has been performed with a full channel estimation each 20:th time. This value is set it to a reasonable high value and the rate of divergence depending on the quantization levels should be analyzed to find out an optimal value and this is a subject for further studies.

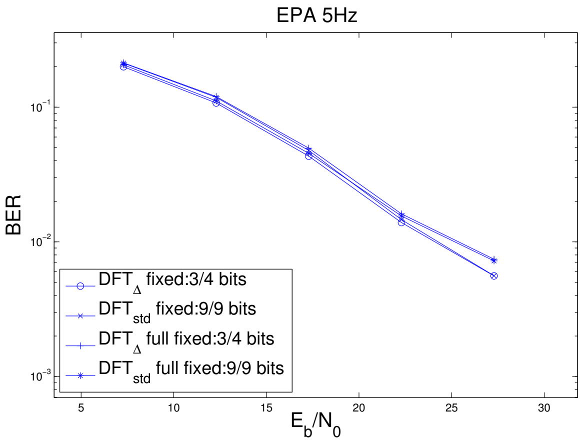

Fig. 7. **Fully fixed and partially fixed implementations**

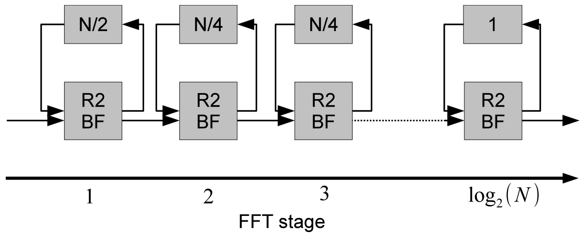

Fig. 8. **Pipelined FFT architecture**

V. Implementation

The pipeline FFT structure is shown in Fig 8. A number of butterflies are connected in a pipelined fashion and each stage has a memory/registers associated with it. The wordlength should increase after each stage in the FFT pipeline to prevent overflow in the butterfly stages. Previous analysis show that the increase of bits can be stopped at an early stage [12] generally described by Fig. 9, where the wordlength increase is stopped at a certain stage, e.g. K=3. The relation between the different parameters is

$$Q_{output} = Q_{input} + P + K - L. \tag{15}$$

Q_{input} is the quantization level of the input data, P is the number of additional bits after the first stage in the FFT, K is the number of FFT stages where the wordlength is increased (by one bit in each stage) and L is the difference in bits between quantization level of the final stage compared to the output quantization level, Q_{output}.

A pipelined FFT structure with normal ordered inputs use memories of different size, see Fig. 8, in each stage. At a certain level the memory overhead needed becomes too large and registers are more advantageous in terms of area consumption. The memory required in the last stages of the pipelined FFT is mapped to registers and the threshold is set to 32 words. Another property in the pipelined FFT structure is that the number of unique twiddle factors decrease in each stage. This enable true constant multiplication in the

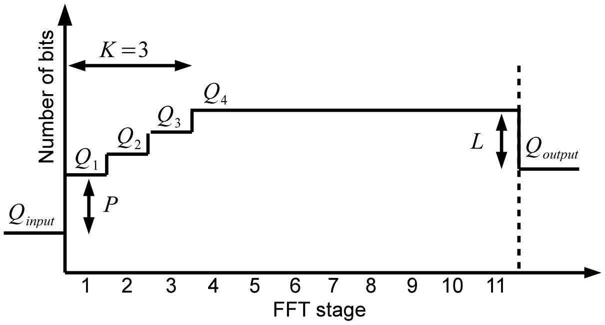

Fig. 9. **Input/output and internal quantization levels of the FFT**

TABLE I
SYNTHESIS RESULTS

	Proposed method	FFT	Song [13]
Technology	65nm	65nm	90nm
FFT size	2048	2048	2048
Freq. (MHz)	100	100	300
Core area (mm^2)	0.16*	0.26	1.16
Normalized** area (mm^2)	0.16	0.26	0.6
Throughput (Msamples/s)	100	100	240
Method	Radix-2	Radix-2	Radix-2
Q_{input}	3	9	9
Q_{output}	4	9	9
P/K/L	1/3/3	1/2/3	-

* The 2048 memory in Fig. 3 is excluded
** Normalization is done through factor: $(65/95)^2$ for 90nm

last stages, realized by shifts and adds in contrast to full multiplication in the cases where there are many option for the twiddle factors. The Catapult tool makes the decision where it is advantageous to use constant multiplicators and where to use full multiplicators.

The proposed method, marked by the dotted line in Fig. 3, and a regular FFT have been synthesized to demonstrate the area reduction. The same FFT architecture is used in both designs and they have been synthesized in a standard 65nm CMOS process from ST Microelectronics. A comparison with an existing FFT (the standard DFT approach) has also been included in Table I to benchmark the existing architecture. The P,K and L values are chosen based on the simulation results in Fig. 7. The clock frequency has been chosen to 100Mhz to approximately correspond to the required processing time for channel estimation operation. The value of the frequency is not the primary concern; rather the clock frequency for the proposed method and the standard method should be kept the same to provide a fair comparison. The design in [13], for the benchmarked architecture does not reveal the internal quantization method or coefficient: P, K and L. The reason for the difference in area might be that the benchmark architecture uses significantly higher internal quantization levels. The reduction in area is about 40% when comparing the standard method with the proposed one. The comparison is done on a subset of the channel estimator, the FFT, other parts of the estimator, such as memories, IFFT and other subblocks are not included. The power consumption is expected to be reduced by approximately the same amount as the area.

VI. CONCLUSION AND FUTURE WORK

This paper has described a fully fixed-point implementation of a new algorithm for channel estimation, operating on channel increments rather than directly on the channel itself. Analytical tools have been used to estimate the potential reduction of wordlength and both simulation and synthesis results demonstrate that the new estimation method achieves considerable gains in area. Around 40% gain in area is observed in the case of a 5Hz Doppler channel and power is expected to be reduced by the same amount. The actual wordlength reduction is consistent with the results from the analytical method. Future work includes power gain estimation and reducing the complexity further by including the IFFT in the feedback path.

VII. ACKNOWLEDGEMENT

The authors would like to thank The Swedish Foundation for Strategic Research and the Marie Curie Actions for providing funding to this research. We would also like to thank Calypto for providing the High Level Synthesis tool: Catapult.

REFERENCES

[1] J. A. C. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," *IEEE, Communications Magazine*, vol. 28, no. 5, pp. 5–14, 1990.

[2] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*, 1st ed. Academic Press, 2011.

[3] 3rd Generation Partnership Project, "Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE," 3GPP, http://www.3gpp.org/, Tech. Rep. 36.888, 2013.

[4] O. Edfors, M. Sandell, J.-J. Van De Beek, S. K. Wilson, and P. O. Börjesson, "Analysis of DFT-based channel estimators for OFDM," *Wireless Personal Communications*, vol. 12, no. 1, pp. 55–70, 2000.

[5] J. Löfgren, L. Liu, O. Edfors, and P. Nilsson, "Improved Matching-Pursuit Implementation for LTE Channel Estimation," pp. 1–12, 2013.

[6] M. Stala, R. Gangarajaiah, O. Edfors, and V. Öwall, "Area and Power Reduction in DFT Based Channel Estimators for OFDM Systems," in *Norchip*, 2013.

[7] 3rd Generation Partnership Project, "3GPP specification: 23.104 V11.5.0; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," Tech. Rep., Jun. 2013. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36104.htm

[8] ——, "3GPP specification: 23.211 V11.3.0; Evolved universal terrestrial radio access (E-UTRA) physical channels and modulation," Tech. Rep., Sep. 2012. [Online]. Available: http://www.3gpp.org/ftp/specs/html-INFO/36211.htm

[9] Jakes, W.C., *Microwave mobile communications*. Wiley, 1974.

[10] M. Garrido, J. Grajal, M. Sanchez, and O. Gustafsson, "Pipelined Radix-2(k) Feedforward FFT Architectures," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 23–32, 2013.

[11] S.-H. Park, D. H. Kim, D.-S. Han, K. S. Lee, S. J. Park, and J. R. Choi, "A 2048 complex point FFT processor for DAB systems," in *International Conference on VLSI and CAD (ICVC)*, 1999, pp. 309–312.

[12] F. Kristensen, P. Nilsson, and A. Olsson, "A generic transmitter for wireless OFDM systems," in *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, vol. 3, 2003, pp. 2234–2238 vol.3.

[13] S.-N. Tang, J.-W. Tsai, and T.-Y. Chang, "A 2.4-GS/s FFT Processor for OFDM-Based WPAN Applications," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 57, no. 6, pp. 451–455, June 2010.