Parameter selection at run-time to optimize energy efficiency

Lodewijk T. Smit, Gerard J.M. Smit and Paul J.M. Havinga

Department of Computer Science,
University of Twente, Enschede, the Netherlands
email:smitl@cs.utwente.nl

Abstract— **Energy efficiency is vital for a mobile terminal. In this paper we investigate how to choose the right parameter settings at run-time so that the energy consumption is minimized while satisfying the required level of service. To use a real world example, the energy consumption of a third generation telephone WCDMA downlink receiver with turbo decoding forward error correction is considered. A trade-off is made between the number of fingers of a rake receiver and the number of iterations of the turbo decoder. A simulation environment is constructed to simulate the system. In this paper we present graphs, with which the trade-off can be easily made.**

Keywords— **wireless communication, quality of service, run-time energy management, rake receiver, turbo decoding, wcdma.**

I. Introduction

MOBILE hand-held multimedia devices evolve in their functionality and complexity with each new generation. As a consequence they require more and more processing power. Another trend is that due to increasing complexity and short time to market, the amount of software in these devices rapidly increases. Low energy consumption is a vital requirement for a mobile device. Only the application of low-power hardware is not sufficient to design an energy efficient mobile, although it is a necessary precondition. Also other aspects have to be taken into account to achieve an architecture with a low energy consumption. In [11] we mentioned: selection of the most suitable algorithms, determination of the most appropriate execution unit, selection of the most optimal parameters and determination of the best power state of different hardware components.

In this paper, we restrict ourselves to parameter selection. With parameter selection, a trade-off is made between different parameter values to select a setting that minimizes the energy consumption of the total system, while still satisfying the given quality of service constraints. This trade-off has to be made at run-time, due to the dynamic nature of the environment of a mobile. Through adaptation of the mobile to the environment, energy usage can be minimized. If these trade-offs were made at design time, the parameters are optimized for worst-case situations, doing too much work in typical situations.

In the next section we will give an overview of trade-off that is made in the selected UMTS case, followed by an overview of the system. Next, our approach is presented how to model the quality and the costs to make a trade-off. In the following section, we look at related work. The next section describes the simulation setup in detail, followed by the simulation results, the conclusions and future work.

II. UMTS receiver example

To investigate the mechanism of parameters selection we defined a practical case within the area of the third generation mobile telephony. In a universal mobile telecommunications system (UMTS) communication system, there are a lot of parameters that influence the achieved quality of service of the wireless link and the energy usage of the mobile terminal (e.g. a phone). This is an ideal case for us, because

- there are a lot of choices to be made,
- there is a lot of dynamic through the changing quality of the wireless link and
- receiving data over a wireless link is computation intensive.

In this paper, we consider the receiver that is needed for the downlink (the base station transmits data to the mobile). We focus on the trade-off between a rake receiver [9] combined with a turbo decoder[4], [5] that are both part of a third generation telephony terminal. A rake receiver improves the quality of the received signal by combining different received reflections of the same transmitted signal. A turbo decoder is a forward error correction (FEC) decoder that tries to correct errors through using redundant information that is added at the transmitting side. The general idea is that there is a trade-off between the rake receiver and the turbo decoder. A very good rake receiver (consuming a lot of energy) gives a good output to the turbo decoder, so the turbo decoder has an easy job to correct the few remaining errors. On the other hand a rake receiver that performs less (consuming also less energy) needs a good turbo decoder (consuming a lot of energy) to obtain the same quality as in the previous case. So our goal is to determine the number of fingers for the rake receiver and the number of turbo decoding iterations with as optimization parameter a low energy consumption, given a certain minimum required quality. However, beside the two mentioned parameters, there are a lot of other parameters which influences the quality as well as the costs. Some parameters (like number of quantization bits, carrier frequency) are determined at design time through definitions in the standard or implementation choices, while others are determined at run time. In our situation, the last category is the most relevant one.

A. Local versus global parameters

A distinction can be made which part of the system determines the values of the parameters. Three different categories are distinguished.

First, some values of parameters are determined by the external environment, e.g. the number of reflections or the number of users. Although the number of users cannot be changed by the system and is not known to the receiver, information about the actual number of users can be very useful to determine the values of other parameters. The base station knows the number of users, so the base station can provide the mobiles with this information.

Second, the transmitter can choose values for parameters (like the spreading factor). The receiver of the mobile can not change this parameters, but by means of a feedback loop it can inform the base station that it has to change the parameters to get a more optimal connection.

Third, the receiver can choose values locally for parameters, like the number of turbo decoding iterations.

This last category of parameter selection - local at the mobile - is investigated in this article. In future work, the scope will be extended to a more global view, including the possibilities at the base station.

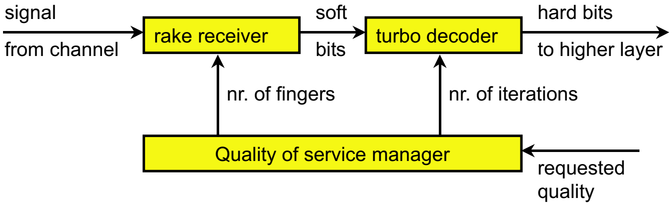

Fig. 1. Combination Of Rake Receiver And Turbo Decoder

III. System Description

Figure 1 shows the combination of a rake receiver and a turbo decoder. The rake receiver is used for wideband code division multiple access systems (WCDMA) [8]. In a WCDMA system all the users transmit in the same band simultaneously. Each user has a unique pseudo random code that is used to identify the specific user on the receiving side by means of correlation performed by the rake receiver. The rake receiver receives signals from different paths with different delays (there are multiple paths due to reflections) and combines them to construct one output signal. Each recognized path element is correlated with a code in a finger to retrieve the original signal. This basic principle of a rake receiver is shown in Figure 2. For a more detailed description, see [8], [13]. The soft output signal of the rake receiver is a signed value. A high positive output value means a high probability that a 1 was transmitted; a low negative output value means a high probability that a 0 was transmitted.

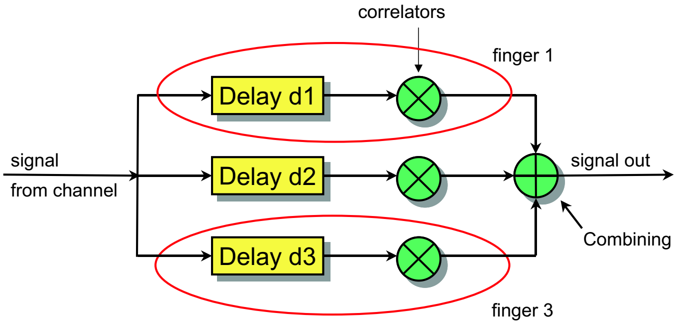

Fig. 2. Basic Principle Of A Rake Receiver

The turbo decoder is an error correcting decoder, which will get the values of the rake receiver as input and produces hard bits (0 or 1) on the output. A block diagram is of the turbo decoder is shown in Figure 3. The turbo decoder is constructed out of two decoders, an interleaver, and a deinterleaver. The turbo decoding algorithm can be run multiple times on the same data to improve the output of the turbo decoder. The output of one decoder is passed to the other decoder, to improve the output by use of a-priori information

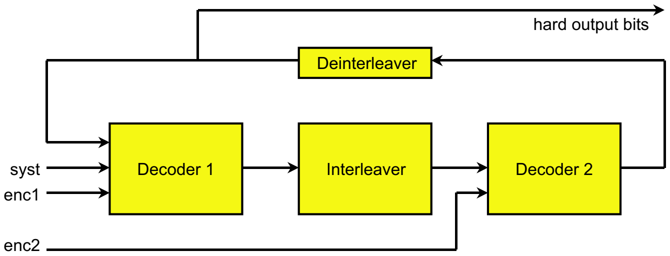

Fig. 3. Block Diagram Of Turbo Decoder

out of the previous iterations. This iterative principle gives the turbo decoder a better performance than a conventional decoder. Between the decoders, the data is (de)interleaved. More details about turbo decoding can be found in e.g. [14]. The Quality of Service (QoS) manager (see Fig. 1) determines the (current) settings for the number of fingers for the rake and the number of iterations that the turbo decoder should do, making a trade-off between the costs and the performance. The minimal requested quality is an input constraint for the QoS manager.

IV. Model

Our goal is to construct a model that minimizes the costs given a set of constraints. This is a typical (non-linear) optimization problem known from operation research [16]. The cost function to be minimized is the sum of the cost function for the rake receiver and the cost function for the turbo decoder. The set of constraints consists of (1) a quality function that must satisfy a minimum threshold for the required minimum quality of service and (2) limits on the range of parameters (e.g. the number of finger must be between 1 and 10). Note that the quality function is not the sum of two quality functions for the rake receiver and the turbo decoder, because the quality of the turbo decoder is dependent on the quality output of the rake receiver. The cost and quality functions will be discussed below in more detail.

A. Cost Function

The energy consumption costs are expressed in number of operations needed for the datapath. This is a rough estimation of the energy costs. The real energy consumption is implementation dependent. Our goal is to make a trade-off between the two, regardless of the exact absolute costs.

A.1 Rake receiver

In essence each finger of a rake receiver performs a correlation of the incoming chips (transmitted pulses) with a code. This means that per correlator sf multiplications have to be done, where sf is the spreadings-factor. These multiplications are very simple, because the polynominal contains only zeros and ones. Therefore, the multiplication is equal to an AND operation. Additional, the results of the different fingers have to be summarized, requiring $sf - 1$ additions (in our implementation). Therefore the number of operations are about $2 \cdot sf$ per processed bit. So, the total number of operations per bit needed for the arithmetic of the rake receiver are about $2 \cdot sf \cdot co \cdot ch$, where sf is the spreading factor (sf is equal to number of chips per bit), co is the number of correlators and ch is the number of channels. Note that correlators are needed for channel estimation and searching as well, so the number of correlators is greater than the number of fingers. Further, most rake receiver designs are more complex (having tracking algorithms, advanced filters, etc.) that make the performance better and the arithmetic complexity higher. In most cases, after the correlation the signal is multiplied with a gain factor before the combining operation, also raising the computation costs. So, the exact costs depends on the design and implementation, but in general the costs per bit are proportional to $2 \cdot sf \cdot co \cdot ch$.

A.2 Turbo decoder

With regard to the turbo decoder, the number of operations per bit is approximately linear to the number of iterations of the turbo-decoding algorithm. There exist different turbo decoding algorithms with different costs and performance. Three frequently used algorithms are Max-Log-MAP, Log-MAP and SOVA. The costs per bit are shown in Table I, taken from [10].

Operation	Max-Log-Map	Log-MAP	SOVA
max ops	5 x 2^M-2	5 x 2^M-2	3(M+1)+2^M)
additions	10 x 2^M+11	15 x 2^M+9	2 x 2^M+8
mul by ±1	8	8	8
bit comps			6(M+1)
look-ups		5 x 2^M-2	
total ops	15 x 2^M-17	25 x 2^M+13	3x2^M+9xM+25
for M=3	103	213	76

TABLE I
Computational complexity of SISO decoders

A turbo decoder contains also two (de)interleavers. The number of operations per bit for the interleavers depends on their implementation. We assume these costs about i. Each iteration the decoding algorithm is executed twice. The number of operations needed for the turbo decoder per bit is about $n \cdot (2 \cdot 103 + 2 \cdot i)$, using the Max-Log-MAP algorithm and an encoder with 3 memories, where n is the number of iterations. The (third generation) Universal Mobile Telecom System (UMTS) always uses three memories in the encoder. So, the costs are linear with the number of executed iterations. Note that these are only the arithmetic costs, e.g. control costs are not included.

The cost function of the system is simply the sum of the cost function of the rake receiver and the cost function of the turbo decoder.

B. Quality Function

There are a lot of parameters that affect the quality (and consequently the costs). However, many parameters are determined by the external environment (e.g. number of paths and interference). These "external" parameters have a considerable influence on the system. Therefore the QoS manager has to take into account the effects of changes of these parameters, but the QoS manager cannot change them. Further, the values of some parameters are determined at or already before design time (e.g. chiprate) and cannot be changed (easily) at run-time. Due to the unpredictable external environment, it turned out to be very difficult to describe the turbo/rake system in an analytical way. The processing costs as a result of the parameter settings can be estimated, but the quality of the output of the turbo/rake system is difficult to determine in advance. To determine this quality of the output for different parameter settings, we constructed a simulation environment. With this simulation environment we can study the effects of different parameter settings. With the results of the simulations, we will try to construct heuristics about what to do in which case. The final goal is to construct a control system that use real-time measurements of the current quality in combination with this heuristics to make the appropriate decisions at run-time.

V. Related work

In the three previous sections we outlined what our goal is, what our motivation is and our approach to achieve this goal. In this section we look at work of others that is related to our work.

The rake receiver was already proposed in 1958 by Price [9]. The related Direct Sequence CDMA (DSCDMA) technique was primarily used for military communications until the late 80's [12]. Due to progress in the development of CDMA techniques, the capacity of CDMA is improved to a comparable level of TDMA and FDMA systems. CDMA is now applicable in cellular systems and therefore the research effort for CDMA systems is increased in the last years.

Turbo decoding was proposed in 1993 [5] by Berrou e.a. After acceptation of the paper [4] of Berrou in the IEEE Transactions on Communications, the turbo code research field became very popular. At this moment, the turbo code theory is transferred to practical applications and implementations.

Therefore in both research fields, WCDMA and turbo decoding, a lot of research has been done already. A lot of research investigates the effect of changing parameters. However, there are several differences with regard to our research.

First, the scope of most of the research is different. Most research specifies on only a part of the system. For example, only the effect of multiple antennas is investigated. Or the capacity of turbo decoding is studied for an AWGN channel. This research is relevant to study the specific characteristics. However, in reality a channel is never a nice AWGN channel. Therefore, we are interested in the combinations of all the parts together.

A second difference is that in most cases the primary research goal is performance optimization. While not neglecting the importance of quality, our primary point of view is optimization for low energy consumption with QoS constraints.

Combining different parts of existing literature is not feasible, because each simulation setup uses different parameters, different assumptions (e.g. different channel models), neglecting sometimes certain issues (e.g. Doppler effect), simplifying the world (e.g. assuming a time invariant environment) and fixing different parameters. These differences make it hard to combine the different results and to draw conclusions for the quality of the overall system. Beside this fact, the amount of data is limited. Most times, only a few plots are presented. Therefore, the only feasible way for our goals is to do our own simulations despite the amount of information available.

VI. Simulation Setup

To simulate the system, we have to simulate the receiver and fec turbo decoder, but also the fec turbo encoder, the transmitter and the channel. In Figure 4 an overview of the system is depicted. Below we will give a short explanation for each different component.

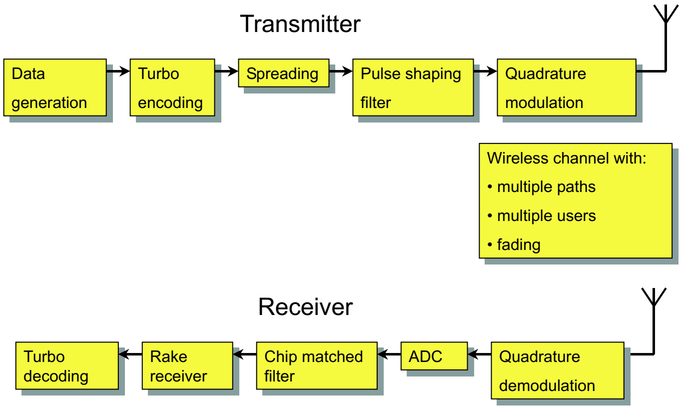

Fig. 4. 3gpp Simulation System

The simulations are performed in Matlab 6.0. For the whole system, a lot of different parameters values have to be chosen. In most cases, we will choose the values that are suggested by the UMTS standard [1]. With the chosen parameters, we think that we simulate a realistic wireless environment, including multiple users, multiple paths, a time-variant fading channel and power degradation. The used parameters are summarized in Table II.

A. Data Generation

The data generator generates blocks with random bit values. The block length may vary over the different simulations. Within the UMTS specification, the block length for the turbo encoder should be between 40 and 5114 bits. The simulations are performed with a block length of 1000 bits.

B. Turbo Encoder

The turbo encoder [2] encodes the information bits. According the UTMS specification, the turbo encoder is a parallel concatenated convolutional coder with two 8-state constituent encoders and one internal interleaver. Trellis termination is a little bit different with respect to the UMTS specification. The first encoder is fully terminated, while the second encoder is left open. The turbo encoder has rate 1/3. So it generates for every incoming information bit a systematic bit, a parity bit of the first encoder and a parity bit of the second encoder. Therefore the length of the output block is three times the length of the incoming block + the number of tail bits (=3). The output bits of the turbo encoder are modulated; 0 and 1 are respectively mapped to -1 and 1. A random interleaver is used.

C. Transmitter

The transmitter spreads the incoming data and generates sf chips for every bit, where sf is the spreadingsfactor. In UMTS, the sf is between 4 and 512. The pn-code used for the spreading is generated with the polynomial $\mathrm{X}^{30}+\mathrm{X}^{23}+\mathrm{X}^{2}+\mathrm{X}^{1}$ for the I branche and $\mathrm{X}^{30}+\mathrm{X}^{29}+\mathrm{X}^{28}+\mathrm{X}^{7}$ for the Q branche. After the spreading, the chips are pulse shaped with a square root raised cosine filter with a roll-off factor of 0.22 and a FIR length of 17. At last the chips are RF modulated using quadrature modulation.

D. Wireless Channel

The channel is modeled as a fading multipath channel with different users that are simultaneous transmitting data. The Doppler frequency is for all simulations fixed to 37 Hz. This corresponds to a velocity of the mobile of 20 km/hour. The receiver gets (in most cases) different reflections of the same transmitted signal that traverse along different paths due to obstacles in the surrounding, like large buildings, mountains, trees and so on. The receiver is not able to separate different reflections that are within the one chip distance of each other. These reflections are considered to belong to the same path. Therefore, the different paths are modeled with delays that are multiples of the chip arrival time. In our model, the second path arrives two chip times later as the first path, the third path arrives three chip times later as the first path and so on. In practice this means that the second path is 150 m longer as the first path and the third path is 225 meter longer as the first path. Due to this longer distance, the reflections with a longer path contain less energy. This is modeled according the general inverse-power propagation law $P_r(r) = Ar^{-\beta}$. The parameter β is very dependent on the surroundings; in our case we choose $\beta = 2$. To compute the average power, we assumed that the distance between the receiver and the base station is 1000 meter. WCDMA is user interference limited [6]. Therefore it is important to model the channel with different simultaneous transmitting users. We made the assumption that the receiver receives all the different users with equal power.

E. Rake Receiver

At the receiver side, the received signal is RF demodulated using a chip-matching filter. Then the received signal is sampled with an over sample factor of 4. Subsequently the data is converted to digital by a digital/analog converter using 6 bits for quantization, which is sufficient according [3]. Next, for each path that is considered, a finger of the rake receiver, see Figure 2, does the correlation. We assumed perfect channel estimation, so that each path is known to the rake receiver.

Tracking of the different paths is done within each finger in the tracking unit. This tracking unit uses a noncoherent Early Late Delay Locked Loop (ELDLL). This time discrete tracking loop samples the signal at the considered perfect time, and a bit earlier and later. With this information the synchronization error is computed to perform the tracking. For the performance of the ELDLL tracking unit see [3]. A disadvantage of this kind of tracking unit is that it is computational intensive, because three correlations are performed for each chip.

F. Turbo Decoder

The turbo decoder consists of two 8-state Soft Input Soft Output (SISO) decoders, separated by a random interleaver. The SISO decoders make use of the LOG MAP algorithm [10]. Note that with regard to the energy consumption, the use of a Max LOG MAP decoder may be a better choice. The performance is worse than the LOG MAP decoder, but it has lower computational complexity, see Table I.

VII. SIMULATION RESULTS

In Figure 5 the simulation results are shown of a simulation with 12 simultaneous transmitting users with a spreadingsfactor of 32, having 3 paths and a rake receiver with one finger. Due to the used forward error turbo coding, the data is transmitted in blocks, which is quite usual for a data connection in general. The block length is 1000 bits. In most data connections, it is not allowed to loose any bit. So, even if only 1 bit out of 1000 bits is wrong after the turbo decoding, the whole block has to be retransmitted. Therefore, we considered that it is important to investigate the frame error rate (FER) as quality measurement instead of the more commonly used bit error rate. In fact, the FER is the information of bit error rate combined with the information of the distribution of the errors over the data.

Figure 5 depicts the Eb/N0 of the wireless channel at the x-axis and the FER at the y-axis. Eleven lines [1] are shown, which connect points at different signal/noise ratio's with the same number of turbo decoding iterations. Simulations are done at the Eb/N0 of [0,1,2,3,4,5,6,7 and 10]. The straight line on top of the graph, with a frame error rate equal to 1, is the quality of the data straight from the rake receiver without turbo decoding. For this line, there are no frames without errors. The line below this line shows the quality of the data after one turbo decoding iteration at different signal/noise rates. Different lines are shown up to a maximum of 10 turbo decoding iterations. Two other graphs are made for respectively two and three fingers that are not shown; they show up the same tendency.

Parameter	value
carrier frequency	$2.0 * 10^9$ Hz
Doppler frequency	37 Hz
chiprate	$3.84 * 10^6$ chips/s
blocklength	40 ... 5112
encoder constraint length	4
Code generator turbo enc.	1 1 0 1 ; 1 1 1 1
turbo encoder rate	1/3
spreading factor	4 ... 512
datarate	chiprate/sf
No. of samples per chip	4
No. of quantization bits	6
roll-off factor SRRC filter	0.22
FIR length SRRC filter	17
modulation	QPSK
number of resolvable paths	variable
number of users	variable
Turbo decoder	LOG MAP
Turbo decoder interleaver	random
No. turbo dec. iterations	variable
No. of fingers	variable
Eb/N0	variable

TABLE II
SUMMARY OF THE SIMULATION PARAMETERS

The first turbo decoding iterations deliver the most gain. After about three or four iterations, the gain is less. To make a trade-off between the number of fingers and the number of iterations is difficult with these graphs. First, for each number of fingers, there is a different graph. These graphs should be combined to make a trade-off. Second, the costs are not included in these graphs. Therefore, these graphs are combined with the costs in new graphs. These graphs depict the relation between the quality and the costs at a certain fixed signal/noise ratio. In Figure 6 such a graph is

---

[1] The lines contain some peaks. This is due to a limited set of blocks that are simulated due to the time that is needed to perform the simulations.

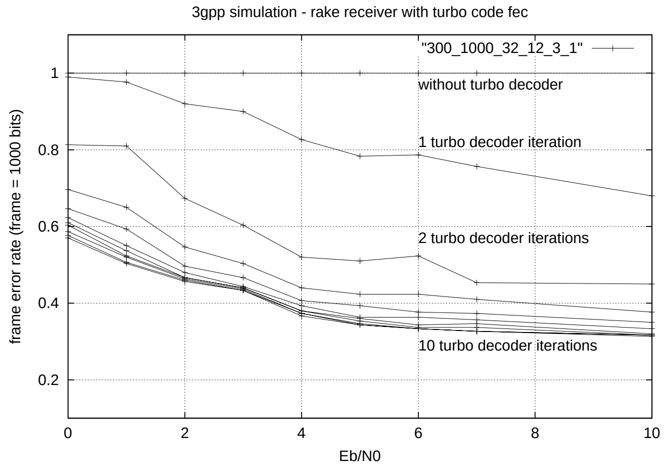

Fig. 5. Different number of turbo decoder iterations with 1000 bits, sf=32, users=12, paths=3, fingers=1

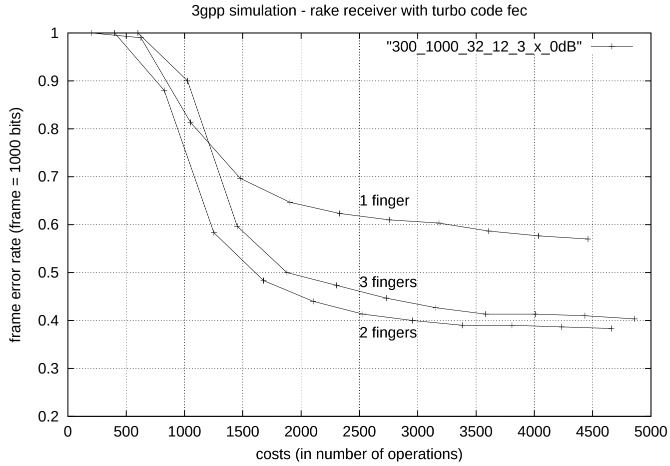

Fig. 6. Relation Between Fingers and Iterations At 0 dB

shown for the same parameters as for Figure 5 at 0 dB with the costs in operations at the x-axis and the quality (FER) at the y-axis. The three lines show the costs for the different fingers. The points are marked with a + sign, which means a measure point after a certain amount of turbo decoding iterations.

The costs in Figure 6 are computed according the formula in section IV-A.1, making the costs $2 \cdot sf$ per bit per finger. Note however that the turbo expands the original information bits to a block with a block-length that is three times the original one. Therefore, the costs are about $6 \cdot sf$ for each original bit per finger. For a sf equal to 32, this means 192 operations per bit per finger. We have computed the graph with 200 operations per bit per finger. With regard to the turbo decoder, the graph is computed with a cost of 426 operations per bit per iteration. We can safely neglect the operations needed for the interleaver as this is only a case of addressing in the right manner. Filtering, which is a computational intensive operation, is not included in the costs because the filtering always should be done, independent of the number of fingers. Further, we do not take into account channel estimation and advanced tracking algorithms. Channel estimation can be done at a relative low frequency, so that the costs are relative low. The ELDLL tracking unit that is used during the simulations is expensive from the viewpoint of energy efficiency and we think that cheaper solutions will satisfy. Also we do not take into account the required control.

In Figure 7 the situation is depicted at the other extreme for for 10 dB.

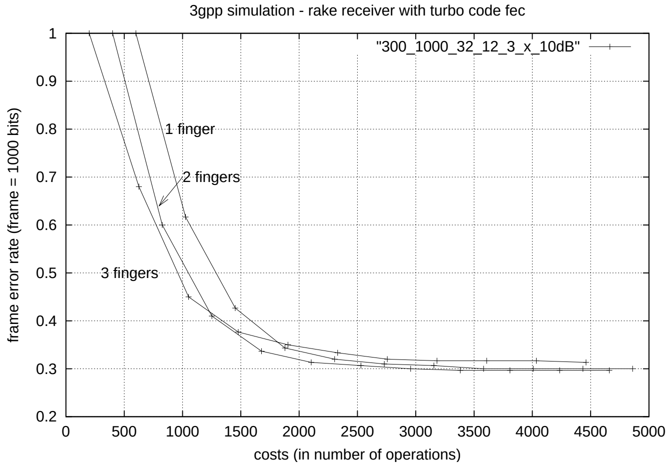

Fig. 7. Relation Between Fingers and Iterations At 10 dB

VIII. Discussion

In this section we will discuss the simulation results. Both in Figure 6 and Figure 7 the quality increase when the number of turbo decoding iterations increases. The gain for each next iteration is less. For the same amount of turbo decoding iterations, the quality is higher (FER lower) for a better signal/noise rate. Clearly a bottom floor in the FER can be seen as the signal/noise ratio increases. At this moment, the user interference (which remains the same) becomes more important as the quality of the channel. This is according the results in [15], see e.g. Figure 7.10 on page 180.

When Figure 6 and 7 are compared, then for the same amount of operations, the quality is better for the 10 dB graph. This is what we expect, because the channel is better.

The quality of two fingers is a little bit better than the quality of three fingers. That is something that we did not expect. We do not have an explanation for this behavior. We suppose that there could be two possible causes for this effect.

First, a limited number of blocks were used in the simulation. The absolute difference between the quality of two and three fingers for 10 turbo decoder iterations at respectively 0 dB and 10 dB is 3 and 6 blocks. The simulation used 300 blocks, so the quality difference between two and three fingers is only 1-2 percent of the total number of blocks. Maybe the simulations use too less blocks. We would like to improve the speed of the simulation environment. At this moment, the simulation of 300 blocks for Figure 5 takes approximately 65 hours to complete on a Pentium 4 PC with 100 percent CPU load.

Second, according to [7], adding number of fingers at higher chip rates may decrease the performance. This can explain the little degradation of the quality from two to three fingers.

Notwithstanding the strange finger effect, these kinds of graphs could be used very well to make the trade-off between the number of fingers and the number of turbo coding iterations. Suppose that the quality must be at least a FER=0.5 and the channel quality is around the 10 dB. From the graph can be seen that the best option is to use three fingers and two turbo decoding iterations. If a FER of 0.35 is the minimum quality, it is better to use two fingers and three turbo decoding iterations.

IX. CONCLUSION

In this paper we presented the first results of a simulation environment for a receiver for an UMTS WCDMA downlink with forward turbo coding error correction. Instead of the common graphs that express performance given a certain channel condition, we constructed graphs that express the amount of energy consumption costs for different scenarios for a given performance. With these graphs, a trade-off can be made in an easy manner about the number of fingers of a rake receiver and the number of turbo decoding iterations that are necessary to achieve the required quality with minimal energy.

X. FUTURE WORK

Clearly, much more parameters are relevant for making a global system trade-off beside the two parameters presented in the previous section. Therefore, more research will be done for the effects of other parameters. For example, a change in the number of users or the blocksize will have a noticeable impact on the performance on the system. Note that there is a difference between the two parameters. The number of users cannot be determined by the system itself and has a noticeable influence on the energy consumption. The blocksize is changeable by the system and does not significantly change the energy costs but will change the latency.

An additional important issue for future work is to include the costs of the transmission of the bits. For example, a higher spreadingsfactor will improve the quality of the received signal but will decrease the transmission speed of the information bits. Therefore the receiver is longer active and requires more energy.

Acknowledgements

This research is conducted within the Chameleon project (TES.5004) supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW. We would like to thank the people from the Turbo Code Research team at the University of Virginia (UVa) for providing the matlab sources for turbo code simulations. Also we would like to thank the people from the institute of microelectronic systems from the technical university of Darmstadt (Germany) for providing the matlab sources for WCDMA simulations. We would like to thank Jos A. Huisken, Kees G.W. Goossens and John T.M.H. Dielissen from Philips Natlab laboratory in Eindhoven for their contributions to this work.

REFERENCES

[1] http:www.3gpp.org.

[2] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network. *3G TS 25.212 v4.1.0 (2001-06): Multiplexing and channel coding (FDD) (Release 4)*, June 2001. see: http://www.3gpp.org.

[3] J. Becker, T. Pionteck, and M. Glesner. Simulation, prototyping and reconfigurable hardware realization of cdma rake-receiver algorithms for flexible mobile transceivers. In *Proc. of ERSA'01*, jun 2001.

[4] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, Oct. 1996.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. International Conference on Communications (ICC)*, pages 1064–1070, May 1993.

[6] T. F. M. Bossert. Interference cancellation in the synchronous downlink of cdma-systems. In *ITG-Fachtaging : Mobile Kommunication*, pages 331–338, Sept. 1995.

[7] D. L. Noneaker and M. B. Pursley. On the chip rate of cdma systems with double selective fading and rake reception. *IEEE Journal on Selected Areas in Communications*, 12(5):853–861, June 1994.

[8] T. Ojanperä. *Wideband CDMA for Third Generation Mobile Communications.* The Artech House Universal Personal Communications Series. Artech House, 1998. ISBN: 0-89006-735-X.

[9] R. Price and P. Green. A communication technique for multipath channels. In *Proceedings of the IRE*, volume 46, pages 555–570, Mar. 1958.

[10] P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. In *Proc. International Conference on Communications (ICC)*, pages 1009–1013, June 1995.

[11] L. T. Smit, G. J. Smit, and P. J. Havinga. Run-time energy management for mobiles. In *proceedings PROGRESS 2000 workshop*, pages 107–112, 2000. ISBN 90-73461-22-7.

[12] M. O. Sunay and P. J. McLane. Snr mismatch and online estimation in turbo decoding. *IEEE Transactions on Communications*, 46(4), Apr. 1998.

[13] G. L. Turin. Introduction to spread-spectrum antimultipath techniques and their application to urban digital radio. *Proc. of the IEEE*, 68(3):328–353, Mar. 1980.

[14] M. C. Valenti. *Iterative Detection and Decoding for Wireless Communications.* PhD thesis, Virginia Polytechnic Institute and State University, July 1999.

[15] P. van Rooyen, M. Lötter, and D. van Wyk. *Space-time processing for CDMA mobile communications.* Kluwer Academic Publishers, 2000. ISBN 0-7923-7759-1.

[16] W. L. Winston. *Operation Research : Applications and algorithms.* Duxbury press, 3 edition, 1994.