Combining Cognitive Radio and Software Radio Approach for Low Complexity Receiver Architecture

Edmund Coersmeier, Klaus Hueske, Marc Hoffmann, Felix Leder, Peter Martini, Harald Bothe

Nokia Research Center, Meesmannstrasse 103, 44807 Bochum, Germany
Edmund.Coersmeier@nokia.com, Marc.Hoffmann@nokia.com, Harald.Bothe@nokia.com
University of Dortmund, Information Processing Lab, Klaus.Hueske@uni-dortmund.de
University of Bonn, Institute of Computer Science IV, Peter.Martini@cs.uni-bonn.de,
leder@informatik.uni-bonn.de

Abstract *– Cognitive Radio techniques will be the next step towards efficient wireless bandwidth utilization. The Cognitive Radio approach is intended to investigate the spectrum allocation and to select an appropriate frequency for the corresponding wireless technology while guaranteeing a highly reliable wireless link. If one extends this approach by balancing between desired high capacity channel and corresponding receiver complexity one can introduce a method for optimum mobile receiver complexity. The support of Cognitive Radio for optimum receiver complexity can only be realized if Software Radio is applied. This paper describes the interaction between Cognitive Radio and Software Radio approach and presents an Artificial Neural Network channel decoder to demonstrate the algorithm complexity flexibility.*

Index Terms *– Cognitive Radio, Software Radio, Multi-Processors, Channel Decoding, Viterbi, Recurrent Neural Networks*

I. Introduction

Cognitive Radio approach is intensively discussed in [1] and intends to efficiently utilize the limited wireless spectrum. Therefore Cognitive Radio scans the spectrum and decides which spectrum or channel, respectively, is the best applicable for the corresponding wireless technology and user application. In [1] Cognitive Radio approach is based on three cognitive tasks, which have been defined as follows: Radio-scene analysis, channel-state estimation and predictive modelling, and finally transmit-power control and dynamic spectrum management. Radio-scene analysis handles interference temperature estimation and detection of spectrum holes on the receiver side. These tasks are not considered in this paper. Instead of channel-state analysis and transmit-power control are important items with regard to potentially optimum receiver complexity.

To realize a good spectrum allocation a high flexibility in the receiver architecture is required. An advantageous approach for such receiver architecture is to implement a Software Radio receiver [2], which is able to choose on the fly one algorithm from a set of different algorithms to solve a specific receiver task like synchronization, channel estimation, demodulation or channel decoding. This new Software Radio approach differs significantly from traditional receiver hardware implementation, because one or several Software Radio processors can run different algorithms based on the same hardware implementation. Instead of traditional pure hardware receiver implementation is realized as an optimal implementation in terms of gate count, silicon costs or power consumption for one or very limited amount of algorithms per receiver task. The drawback of that approach is the missing flexibility to redefine the functionality after implementation has been finalized. Thus pure hardware realization is not supporting Cognitive Radio approach very well. To achieve high data rates with Software Radio the most promising approach is actually to implement a multi-processor platform, which achieves the required processing power and speed by parallel execution of many operations and several algorithms.

Based on multi-processor Software Radio approach this paper investigates how multi-processor platforms can interact with Cognitive Radio to achieve optimum receiver complexity. Optimum means in this context balancing between receiver algorithm processing amount and required receiver performance. Typically pure hardware implementations are designed for worst case transmission scenarios by always applying high

performance receiver algorithms, which fulfil at all times the required receiver performance. This approach is far from optimum in terms of low receiver complexity because the receiver always processes data as it is required for worst case scenarios.

Now this paper proposes to design algorithms in a different way, optimally fitting on parallel-processor platforms to achieve exactly the required algorithm performance, which has been requested from Cognitive Radio. When operating with the exact matching receiver performance then the receiver complexity will be optimally as well.

The paper first summarizes important Cognitive Radio features. This is followed by a description about interaction between Cognitive Radio and Software Radio. After that an overview about potentially multi-processor Software Radio platform principles is given. Finally the new Cognitive Radio approach is analyzed by an example for channel decoding. Channel decoding algorithm and corresponding simulation results are based on Recurrent Neural Network (RNN) seen as alternative to Viterbi decoder.

II. Important Cognitive Radio features

In this section a short summary about channel-state estimation and transmit-power control, as proposed in [1], is given. These two items build the base to develop new meaningful interaction between Cognitive and Software Radio.

A. Channel-State Estimation

Channel-state estimation is an important activity within the Cognitive Radio approach to be able to judge whether a specific channel is able to provide enough channel capacity for the desired transmission technology or user application, respectively. For Cognitive Radio technology a continuous pilot transmission is not seen as reasonable way to analyze the channel conditions because it is wasteful in both, transmit power and channel bandwidth [1]. Instead of [1], [3] propose the use of semi-blind training which first employs a supervised training mode and finally tracks the initial channel state estimation to detect changes in the channel properties, Figure 1.

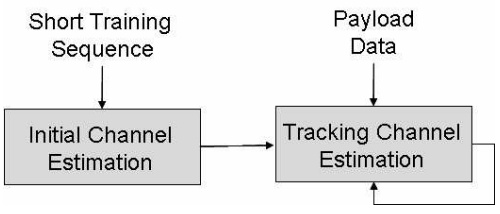

Figure 1 Combining supervised training and tracking mode.

After channel-state information is available in the receiver a so called rate-feedback [1] is provided from the receiver to the transmitter for setting up the data rate as well as the transmit-power control.

B. Transmit-Power Control

Cognitive Radio transmitter gets from the corresponding receiver a rate-feedback. The rate feedback is used to start planning the required transmit-power per transmitter. In a multi-transmitter scenario each data transmission needs to achieve its target data rate and therefore the corresponding transmission power needs to be regulated. In [1] a two-loop setup is proposed, Figure 2.

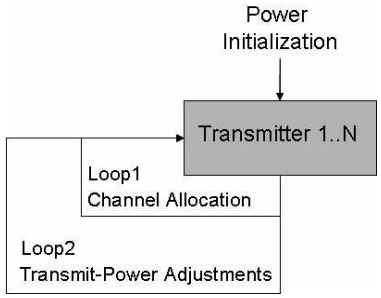

Figure 2 Two-loop setup for bandwidth allocation and transmit-power control.

After initial, equal power setup for all transmitters, a first inner loop, here called Loop1, see Figure 2, iterates from one transmitter to the next, where each transmitter allocates a number of channels based on Water-Filling approach [1]. After bandwidth allocation has been finalized, all transmission systems are investigated in the outer loop, here called Loop2, see Figure 2, whether their actual data rates are exceeding, matching or undershooting the target data rates. In case of exceeding or undershooting the target data rates, Loop2 adjusts the transmit-power of each transmitter so that the actual data rates are matching with the desired target data rates.

After summarizing in this section the proposal for Cognitive Radio from [1] the next section introduces an enhancement to Cognitive Radio approach for optimal mobile receiver complexity.

III. Optimal Receiver Complexity

The actual idea of Cognitive Radio is the optimal utilization of the available wireless spectrum by employing a highly flexible system, which adjusts bandwidth allocation as well as transmit-power to the corresponding requirements to achieve the desired target data rates. From the high-level wireless system perspective this approach is looking comprehensive and

is a significant step forward compared to the actual wireless transmission systems. From the wireless, mobile receiver perspective this approach can still be enhanced by additionally introducing a cognitive complexity regulation for mobile receivers. This section will first describe how to optimally utilize Software Radio for the required receiver flexibility and after that introduces an extension to the Cogntive Radio approach.

A. Software Radio

Figure 3 describes an analysis of Software Radio processor load for an OFDM receiver, completely running on a floating-point DSP processor [4], [5], [6].

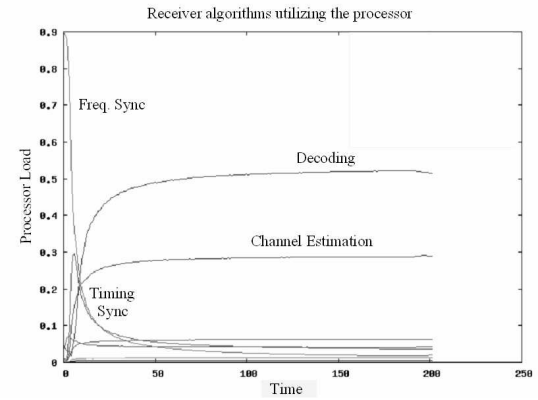

Figure 3 Example for Software Radio processor load for one Digital Radio Mondiale OFDM receiver.

There are a few algorithms which require significant amount of processor load during signal reception whereas several other algorithms individually do not add further large amount of processor load. For example, at the beginning of the receiver activities in Figure 3 synchronization algorithms for frequency and time synchronization require most processing amount whereas during tracking mode channel estimation as well as decoding algorithms significantly contribute to the overall processor load.

Transferring these results to the Cognitive Radio approach it is valuable to enhance the cognitive approach to support the employment of low complexity algorithms for special set of receiver tasks if possible. In some cases the support could be long-lasting in other scenarios one can assume temporary support from the Cognitive Radio approach to allow the mobile receiver to employ low complexity algorithms.

The motivation for low complexity algorithms can be described with the following items:

- Software Radio processor overload
- Receiver power consumption reduction

1) Software Radio Processor Overload

Software Radio processor overload can be the case if one runs more than one radio receiver at a time on a single or multi-processor field. Depending on each receiver complexity there might be temporary shortage of available processor power.

2) Receiver Power Consumption Reduction

Receiver power consumption reduction could be introduced as an option for the mobile user. If the user selects to run a very low power device, the handheld should be smart and understand how to reduce power consumption as much as possible. This could be done by informing the radio receiver to be as power efficient as possible.

Typically traditional receiver designs balance algorithm complexity and algorithm performance before implementation. Hardware implementations target high algorithm performance to handle worst case transmission scenarios, e.g. bad channel conditions. In case of good channel conditions, the hardware receiver is over-specified from the performance perspective. Unnecessary receiver battery power is wasted. This problem can be circumvented when employing a Software Radio receiver to find a compromise between algorithm performance and power consumption.

3) Software Radio Flexibility

Software Radio flexibility offers the chance to load different algorithm complexities for the same receiver task, depending on the channel conditions. In case of difficult channel conditions, high performance algorithms, requiring high processor load, can be applied. If channel conditions are not that worse, Software Radio receiver can reload an algorithm set with lower performance and less processor load. An example is shown in Figure 4 by introducing three algorithms sets.

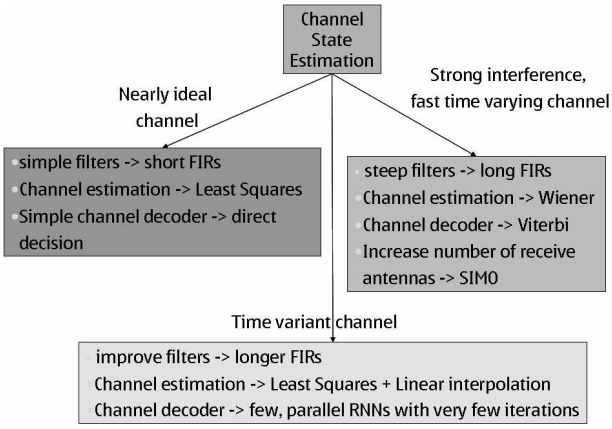

Figure 4 Software Radio approach to load receiver algorithms with different processing complexity.

The channel state estimation selects one of three sets of receiver algorithms, depending on the properties of the channel conditions, identified in the channel state estimation process. Nearly ideal channel conditions lead to simple receiver algorithms, which do not fully load single or multi-processor platform. When channel conditions get more critical, in Figure 4 defined as time variant channel, more complex algorithms have to be employed. This setup will require already significantly more load than the first algorithm set. Finally the worst case scenario considers strong interference and fast time varying channel conditions and the channel state estimation process selects the algorithms for highest receiver performance.

Figure 4 acts as Single-Input-Single-Output (SISO) channel example, which could be expanded for Multiple-Input-Multiple-Output (MIMO) cases. It is imaginable to select a subgroup from each algorithm set instead of selecting always a complete new set of algorithms as function of channel conditions.

After introduction how Software Radio can react on different channel conditions, the next section introduces the extension to the Cognitive Radio to enable selection of corresponding algorithm sets.

B. Cognitive Radio Enhancement

Cognitive Radio approach from [1] can be enhanced to a three-loop approach to introduce receiver complexity optimization. Figure 5 presents the modified drawing from Figure 2 by adding one additional outer loop, called Loop3.

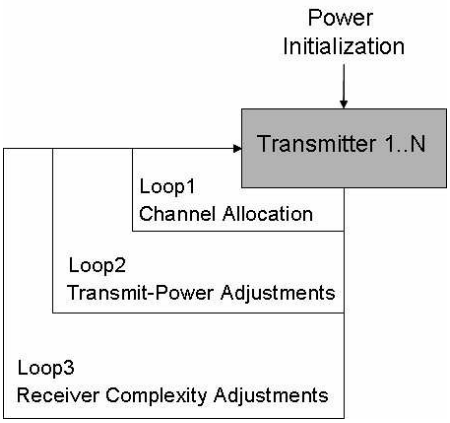

Figure 5 Three-loop extension for Cognitive Radio to support receiver algorithm set selection.

Figure 5 shows that after power initialization for all transmitters the channel allocation process starts and is lead over to transmit-power control. After Loop1 and Loop2 have been finalized, all transmitters run from data-rate perspective with optimal transmit-power. Now a third loop can be activated, which checks from all receivers an optional request for low complexity receiver algorithms. For all receivers, which indicate the wish for low complexity algorithms, a selection, based on Figure 4, starts. Selection process incorporates both previous loops, Loop1 and Loop2, respectively. Two system properties need to be investigated to enable the requested reduction of receiver complexity for selected radios:

- Transmit-Power Increase
- High Quality Channel Selection

1) Transmit-Power Increase

If within Loop2 the transmit-power of the corresponding transmitter is increased, less complex receiver architecture can be installed for that corresponding radio. At the same moment other radio receivers might need to increase a little bit the own receiver complexity because their transmitters might reduce the transmit power. But all those transmit-power reductions should be very small amount for each transmitter because absolute required power reduction will be distributed as a fraction over all transceivers. Thus receiver complexity reduction through transmit-power increase should be only considered as one minor possible option. If all radio receivers would always request power reduction, the system finally might not come to an improvement for any receiver when regulation is done via transmit-power increase.

2) High Quality Channel Selection

As a better alternative to transmit-power increase the channel selection process within Loop1 can be reactivated. If the Cognitive Radio manages to find another, better fitting, free high quality channel for the corresponding receiver, the low complexity request can be solved without transmit-power changes for any transmitter. If no further free channels are available it might be possible to exchange the already allocated channels of two radios, one radio with low complexity request, the other radio without activating this request option. Thus Loop1 channel re-allocation procedure might be the better option from the overall system perspective, because less action is required from all Cognitive Radios.

3) Low Complexity Parameter for Loop1 and Loop2

Another alternative is to leave out Loop3 at all and to consider already during first run of Loop1 and Loop2 the low complexity option. Each channel selection process and each transmit-power adjustment would be informed beforehand whether the corresponding receiver asks for low complexity support. If the low complexity option is enabled by one or more receivers, Loop1 and Loop2 take one more parameter into account

for the overall selection process. Thus the selection process is more complex during each run of Loop1 and Loop2, respectively, but this principle removes the need for Loop3 at all.

After investigation of the new Cognitive and Software Radio approach has been finalized, next sections introduce practical example how flexible channel decoder approach can implement the new idea. Therefore, first a short overview about possible Software Radio platforms will be given. After that Recurrent Neural Network channel decoder will be described and simulation results will be presented.

IV. Multi-Processor Software Radio Architecture

To successfully replace in future some pure hardware radio implementations by establishing new Software Radio designs, it is required to implement a high performance processor platform, which runs one or more radios in parallel at the same time. Power consumption is an important parameter for embedded systems hence it is not possible to speed up processors until the required processing power is available. Instead of multi-processor platforms seem to be a more realistic approach to address Software Radio requirements. Basically one can differentiate between heterogeneous and homogeneous multi-processor platforms, as shown in Figure 6.

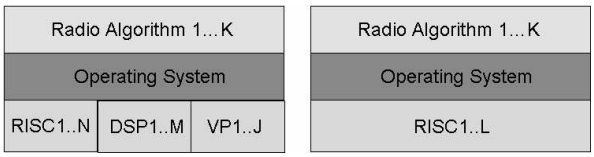

Figure 6 Heterogeneous and homogeneous multi-processor platform for Software Radio.

The left drawing in Figure 6 shows a heterogeneous platform, including different types of processors, e.g. RISC, DSP and Vector-processors [7]. The right drawing consists out of L processors, whereas only one processor type, e.g. RISC processor, has been implemented, see [8]. The Operating System (OS) bridges between hardware and radio algorithms. The algorithms are implemented processor independent, except for word length limitations, because C-Code instead of Assembler implementation is favoured. This approach makes software implementations better reusable without completely re-writing code in case of changes in hardware. New compilation run updates the object codes and executables. On the other hand abstract C-coding might be less efficient from performance perspective than implementing Assembler.

The OS scheduler assigns algorithms at runt-time to the corresponding processors. This task is in case of heterogeneous platforms more difficult than for homogeneous platform. When new algorithms need to be activated for the next radio task, both platform types need to identify and select an actually free processor. In case of heterogeneous platform an additional decision about best fitting processor type needs to be made. This additional selection process assumes that for one algorithm different executables, corresponding to the different processor types, are available.

If utilizing a multi-processor platform for Software Radio approach it is important to implement the radio algorithms in such way, that parallel processing is supported by the algorithm architecture. High performance Software Radio algorithm architectures might significantly differ from parallel hardware implementations because implementation bottlenecks like memory access, BUS bandwidth as well as available processing power are different for ASIC / FPGA designs and processor systems. When planning parallelization of algorithm parts, it needs to be considered that granularity of each algorithm segment should be high for Software Radio algorithms. The reason for that can be found in the limit of possible data transfers in the Software Radio platform. The longer one algorithm segment can work on a set of data the more efficient is the Software Radio platform utilization. For pure hardware implementations this statement needs not to be true, because data exchange can be speed up by direct wiring.

The next section investigates a channel decoder algorithm, which provides good decoding performance and can be parallelized on high abstraction C-code level.

V. Parallelization of Channel Decoder

Typically channel decoding of convolutional codes is done via Viterbi algorithm [9], [10] but can also be done by Recurrent Neural Networks (RNN) [11], [12]. The advantage of RNN is the flexible adaptation of algorithm parameters to adjust processing speed, algorithm performance as well as parallelization on high abstract level. Thus RNN channel decoder is a good candidate to fit on multi-processor Software Radio platform.

The encoding of information bit vector $\mathbf{a}$ to code word $\mathbf{c}$ is done by multiplication with generator matrix $\mathbf{G}$.

$$\mathbf{c} = \mathbf{G}^T \cdot \mathbf{a} \tag{1}$$

Interpreting the channel decoding process at the receiver as minimum search procedure, one can

compare the RNN approach with well known Least-Squares approach from adaptive filter theory [12], whereas error e is build from receive matrix $\mathbf{X}$, adaptive filter vector $\mathbf{w}$ and reference signal $\tilde{y}$. Variable n denotes time instances.

$$\min_{\mathbf{w}(n)} \left\| e(n) \right\|_2^2 = \min_{\mathbf{w}(n)} \left\| \mathbf{X}(n)\mathbf{w}(n) - \tilde{y}(n) \right\|_2^2 \tag{2}$$

The corresponding channel decoding problem is based on error $\mathbf{e}$, receive vector $\mathbf{r}$, code word $\mathbf{c}$, generator matrix $\mathbf{G}$ and information bits $\mathbf{a}$.

$$\min_{\mathbf{c}} \left\| \mathbf{e} \right\|_2^2 = \min_{\mathbf{c}} \left\| \mathbf{r} - \mathbf{c} \right\|_2^2 = \min_{\mathbf{a}} \left\| \mathbf{r} - \mathbf{G}^T \mathbf{a} \right\|_2^2 \tag{3}$$

Equation (1) can be modified to

$$c_k^{(j)} = -\prod_{i=1}^{K} \left(- a_{k+1-i} \right)^{g_i^{(j)}} \tag{4}$$

In equation (4) $c_k^{(j)}$ describes the kth coded bit at jth encoder output. The exponent $g_k^{(j)}$ is the ith component of the impulse response from the jth output. With an example of code rate $1/n = 1/2$ and constraint length $K = 3$ the error function can be described as

$$\min_{\mathbf{a}} f(\mathbf{a}) = \min_{\mathbf{a}} \sum_{k=0}^{l-1} \left| r_k^{(1)} + a_k a_{k-2} \right|^2 + \sum_{k=0}^{l-1} \left| r_k^{(2)} + a_k a_{k-1} a_{k-2} \right|^2 \tag{5}$$

Variable l denotes the length of the information bit vector. The characteristic of the error surface depends on the noisy input vector $\mathbf{r}$ and is assumed to be constant. Vector $\mathbf{a}$ needs to be defined by the minimization process in that way that the error-function will be as small as possible. The minimization process can be arranged by gradient approach.

$$grad(f(\mathbf{a})) = \nabla f(\mathbf{a}) \tag{6}$$

Parallelization of the network can be done by starting different decoders in parallel, each with another initialization starting point for the gradient. After all decoder gradients have finalized their iterations, a final selection process evaluates the most probable winner candidate. The setup is shown in Figure 7.

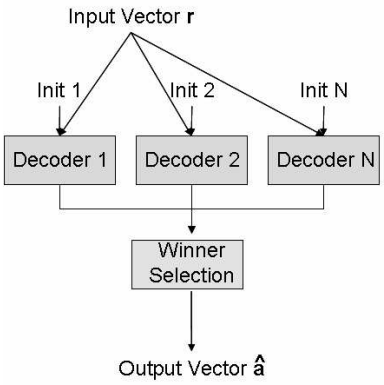

Figure 7 Winner selection from N parallel decoders.

Each of the N parallel channel decoders iterates on the same receiver data set. Thus no data transfer between the neural networks during gradient search process is required and consequently each decoder can be executed on an own processor from the multi-processor Software Radio environment.

Based on that RNN approach the next section presents simulation results to show how well the RNN performance of parallel decoders is compared to traditional Viterbi decoder and direct decision approach.

VI. Simulation Results of Parallel RNN Decoders

First starting point is traditional decoder analysis. Working point of 1e-4 BER is assumed. In Figure 8 Viterbi and direct decision decoders have a significant difference of 6.5dB performance.

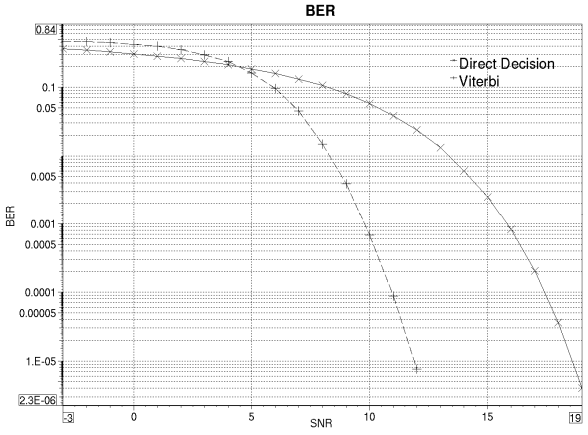

Figure 8 Comparing traditional Viterbi and direct decision decoders.

The huge performance discrepancy is based on significant difference in complexity for both algorithms. Thus employing both algorithms in a receiver design gives not much freedom when trying to

reduce complexity with performance degradation in small steps.

When introducing RNN approach from previous chapter as an alternative to Viterbi decoder, it is important to investigate how much differences in performance appear between Viterbi and RNN solutions.

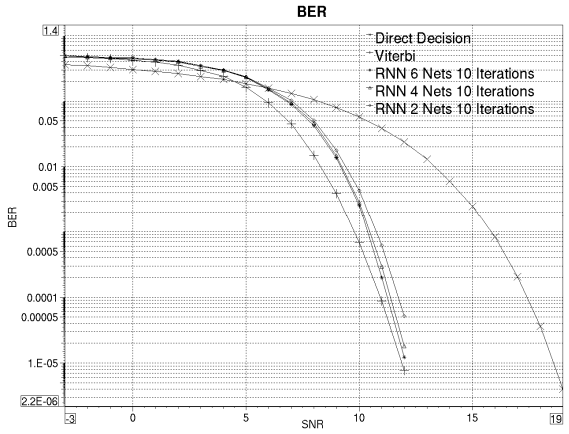

Figure 9 Six, four and two parallel RNNs.

Figure 9 shows three additional RNN decoders running 6, 4 and 2 RNN subnets in parallel, respectively. Each network's gradient makes 10 iterations until it decides the most probable received bit. At working point 1e-4 four parallel RNN subnets (RNN 4) gain 0.35dB over two parallel subnets (RNN 2), six subnets (RNN 6) gain 0.15dB over four subnets (RNN 4). Finally the Viterbi algorithm gains 0.3dB over six subnets (RNN 6). This can be seen in detail from Figure 10.

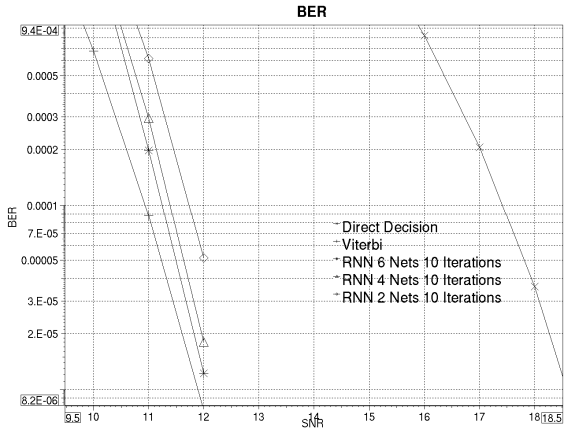

Figure 10 Zooming into working point 1e-04 BER.

Although Viterbi cannot be passed by RNN approach from performance perspective, it is obvious that RNN provides an important alternative for configurable complexity receivers. The RNN setup provides a highly parallel architecture, where any number of networks reaches significant better performance than direct decision decoding.

VII. Conclusion

This paper introduces a novel extension to Cognitive Radio approach by combining optimal channel selection as well as transmit-power control with flexible receiver architecture definition for Software Radio implementation. Balancing between receiver complexity and receiver performance can be achieved in excellent way when employing Recurrent-Neural-Networks for convolutional channel decoding. The RNN architecture can be implemented with parallel subnets, which can nicely fit into a multi-processor Software Radio platform. When reducing the number of RNN subnets and with that the receiver complexity, the radio performance degradation is small compared to direct decision algorithm.

References

[1] Haykin S., Cognitive Radio: Brain-Empowered Wireless Communications, IEEE Journal on Selected Areas in Communications, Vol. 23, No. 2, February 2005

[2] Jondral Friedrich K., Software-Defined Radio—Basics and Evolution to Cognitive Radio, EURASIP Journal on Wireless Communications and Networking 2005:3, 275–283

[3] Haykin S., Huber K., Chen Z., Bayesian sequential state estimation for MIMO wireless communication, Proc. IEEE, vol 92, no. 3, pp439-454, March 2004

[4] Coersmeier E., Bauer A., Kosakowski M., Pfuhl N., Spahl B., Tolksdorf D., Xu Y., Gerharz M., Hansmann W., Leder F., Evaluation Architecture for Digital Radio Mondiale Multimedia Applications, CSA2005, Banff, Canada, July 2005

[5] Coersmeier E., Bauer A., Saez C., Hotz A., Hoffmann M., Kosakowski M., Xu Y., Dynamic Wiener Filter Coefficient Update for Software Radio, Workshop on Software Radio, Karlsruhe, Germany, 2006

[6] Coersmeier E., Bauer A., Saez C., Hotz A., Hoffmann M., Kosakowski M., Xu Y., Improving Channel Estimation for Software Radios, CSA 2006, Banff, Canada, July 2006

[7] Schwoerer L., Moermann K., Benchmarking MIMO OFDM Algorithms on the EVP, GSPx – the 4th International Global Signal Processing Conference, Santa Clara, California, October 2006

[8] Goodacre, J.; Sloss, A.N., Parallelism and the ARM instruction set architecture, IEEE Computer Society, Volume 38, Issue 7, July 2005 Page(s):42 - 50

[9] Viterbi A.J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Technology, 13:260-269, April 1967

[10] Forney G.D., The Viterbi Algorithm, Proc. IEEE, 61: 268-278, March 1973

[11] Hämäläinen A., Henriksson J., Convolutional Decoding Using Recurrent Neural Networks, Procededings of Inter. Joint Conf. on Neural Networks, 5:3323-3327, San Diego, 1999

[12] Hueske K., Reduzierung der Komplexität von Verfahren zur Kanaldecodierung, Diplomarbeit D2-06, University of Dortmund, Feb. 2006

[13] Haykin S., Adaptive Filter Theory, Prentice Hall, Third Edition, 1996