CHAPTER Eight

Conclusions and Recommendations

34 Introduction

Receiver sensitivity is one of the key specifications and important attribute of any mobile communication receiver. The intension of this research work was to explore and find out different possible options to improve the receiver’s sensitivity level. There are several design techniques explored and some new methods are found during this research work for improving the mobile receiver sensitivity level. Some techniques are complex, some are simple, and some are costly, whereas some are cost effective. So, optimizing the communications receiver design is inherently a process of trade-off and compromise. In this research work, all the potential design areas are explored and the corresponding optimum design solutions as well as new techniques are found and recommended.

35 Adaptive switching based receiver architecture for optimum receiver performance

Some of these identified methods are very complex and requires lot of processing power, which leads to more delay and battery power consumption. On the other hand, some of the algorithms provides best performance in a specific channel conditions and does not give much performance benefit in other conditions. So, there is a need of adaptive
switching mechanism to achieve the optimum performance of a receiver. The switching will adaptively enable or disable different receiver processing blocks/algorithms on the basis of detected channel conditions, received signal strength, and detected error to achieve the maximum performance with minimum processing power.

Here, in this receiver architecture four switches are proposed to adaptively switch based on the scenarios and requirements, which lead to optimum performance with minimum power consumption. The switches are enabled or disabled based on the detected environments and system power, performance trade-off metric. The adaptive switching based receiver for optimum power-performance is shown in the figure 8.1. This uses all the explored new techniques for sensitivity and receiver performance improvement and at the same time it employs some switches at every point to take an adaptive decision based on the power-performance trade-off.

(A) **Re-encoded Soft-bit generation from decoded bits based on soft-bit quality and CRC status (Switch_2):**

The switch_2 is placed after the CRC checking unit and it takes CRC status and Viterbi decoder produced soft frame quality estimates as an input. The CRC checking unit indicates whether the CRC of the decoded bits is passed or failed. On the other hand the soft frame quality (SFQ) estimator estimates the received soft bit quality during the Viterbi decoding process and indicates the quality value. The estimation of SFQ as a part of Viterbi decoder process is already discussed in chapter 5, “method for soft frame quality and raw BER estimation from Viterbi Decoder” and patent application is filled on this. That is a very simple technique and estimates the frame soft quality without any computational overhead. Now, the switch_2 evaluates the CRC and SFQ status and if the
CRC status is failed and SFQ value is above a threshold value (the higher the SFQ value lower the frame or block quality), then that indicates that most of the decoded bits are wrong, so this decoded bits cannot be used for iteration or 2nd time receiver processing for better combined soft-bits generation or channel re-estimation. So, the process is given up here and the FER or BLER is marked as failure and no iteration process is triggered. Because, if the generated soft bits (S1) are too bad in quality, then the decoder can’t correct those bits, rather it might deteriorate the situation further by going to wrong path in the Trellis structure and leading to wrong output bits. That detection and switching disable decision saves lot of processing power and time for 2nd time iteration.

On the other hand, if the CRC status is failed and SFQ value is above a threshold value, then switch_2 is enabled and the iterative receiver path is enabled for better combined soft-bits generation or channel re-estimation.

If CRC status is pass, then there is no need of iteration as the frame decoding is already successful, so the switch_2 is always disabled as long as the CRC status is indicated as pass. That will help to save processing power and time. The switch_2 decision logic is shown in the table-8.1.

Table-8.1, Switch_2 decision logic

<table>
<thead>
<tr>
<th>Switch_2 Input status</th>
<th>Switch_2 enable or disable decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC status: Fail &amp;&amp; SFQ: greater than threshold</td>
<td>disable</td>
</tr>
<tr>
<td>CRC status: Fail &amp;&amp; SFQ: lower than threshold</td>
<td>enable</td>
</tr>
<tr>
<td>CRC status: Pass &amp;&amp; SFQ: greater than threshold</td>
<td>enable</td>
</tr>
<tr>
<td>CRC status: Pass &amp;&amp; SFQ: lower than threshold</td>
<td>enable</td>
</tr>
</tbody>
</table>
(B) Enable iterative channel estimation unit based on estimated burst quality, CRC status and CCI/ACI/AWGN channel type detection (Switch_3):

If the switch_2 is enabled then that indicates that though the CRC is failed but still these generated hard bits can be re-used for iterative channel estimation process to improve the channel estimate. This is because though the CRC fails but the convolution decoder corrects many of the input bits (soft bits) due to decoding gain [1, 2, 4]. So, these decoded bits are first re-encoded, punctured according the encoding and puncturing schemes used in the transmitter side. Then these data bits are interleaved according to interleaved pattern used in the transmitter side. Next these data bits are converted to soft bits (S2) by converting 1 to –ve Max and 0 to +ve Max value (Max value could be 0.333 or any other value as used in the receiver to represent the maximum scaled soft values). Next the burst quality is estimated comparing the sign of first time generated soft bits (S1) with the re-encoded soft bits (S2).

Switch_3 enabling/disabling decision:

The input to the switch_3 are: estimated burst quality [Weight_burst_1, …., Weight_burst_m], detected channel type ACI/CCI/AWGN dominated channel, and switch_2 status.

(a) The burst quality (Weight_burst_1, …., Weight_burst_m) are estimated using the above equation (1). If the estimated burst quality for a burst is above a threshold value then this is considered for the re-estimation process. That means the condition is Weight_burst_m >= Threshold for enabling the re-estimation of that burst.

(b) The ACI/CCI/AWGN detector: The re-estimation of channel is more effective and found better when the channel type is AWGN limited e.g. the received signal is
dominated by noise signal. Earlier one method was proposed for ACI/CCI/AWGN channel type detection and the details about that patent application filed [33] - “P36911_US1_2012-07-05_application_as_filed_0111-141- Method for interference and carrier power estimation and its application to automatic gain control and SNR computation”. This method helps to find out the channel type - that means whether the channel is AWGN or ACI or CCI type. Now, if the channel type detected is AWGN then this re-estimation method is applied.

(c) Switch_2 status: The status of the switch_2 should be enabled for switch_3 to operate.

The detail about the switching strategy of switch_3 is given in table-8.2.

Table-8.2, switch_3 switching criteria

<table>
<thead>
<tr>
<th>Switch_3 Input status</th>
<th>Switch_3 enable/disable decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch_2 status: enable</td>
<td>enable</td>
</tr>
<tr>
<td>Burst_m quality: greater than threshold</td>
<td></td>
</tr>
<tr>
<td>Channel type: AWGN</td>
<td></td>
</tr>
<tr>
<td>Else for any other conditions</td>
<td>disable</td>
</tr>
</tbody>
</table>

This above conditions helps to enable the iterative channel estimation when it will provide gain, and in rest of the conditions where there will be no gain, it is disabled so that the processing power consumption is avoided when it does not provide performance gain.

(C) Enable Angle Demodulator based on detected channel propagation type and estimated C/I: (Switch_4):

From this research work, earlier one patent filed “P34845US1_2011-10-16- Systems and methods for demodulating a signal” [30]. In that proposal, the quality of the soft bits is improved by using diversity combining by demodulating the soft values through the
conventional legacy receiver based on DFSE or MLSE or MAP equalizer along with a very simple, low complexity angle difference based demodulator. Here, the basic principle is demodulating the information in parallel with multi-model approach (one demodulation per model). The simple angle based de-modulator exploits the modulation properties and demodulates the bits. Here, as the soft bits are generated without channel estimation process, so, the demodulated bits are independent of channel estimation. But, this method will work well when the channel type is noise limited e.g. AWGN type.

The switch_4 is enabled, when the CRC status is failed, detected channel type is AWGN and estimated channel length is short (e.g. 3 or 4 tap).

(1) CRC status is detected from CRC checking unit.

(2) Detected channel type: The channel type is detected using the method as discussed earlier.

(3) Channel Length Detection: For detecting the channel length (delayed or non-delayed channel), these below steps will be performed:

(i) First perform a 4-tap channel estimation and then perform a 7-tap channel estimation to find the scaled squared error in each case; where it’s computed as, 
\[
\text{Scaled}_\text{sqerror}_4 = 4\text{-tap channel estimation error } \times \text{ AIC factor.}
\]
\[
\text{Scaled}_\text{sqerror}_7 = 7\text{-tap channel estimation error } \times \text{ AIC factor.}
\]

(ii) Then the winning channel length is decided as 4-tap or 7-tap based on the obtained minimum squared error from these two Scaled_sqerror values.
\[
\text{Channel_length} = \text{4-tap if } \text{Scaled}_\text{sqerror}_4 = \min(\text{Scaled}_\text{sqerror}_4, \text{Scaled}_\text{sqerror}_7).
\]
\[
\text{Channel_length} = \text{7-tap if } \text{Scaled}_\text{sqerror}_7 = \min(\text{Scaled}_\text{sqerror}_4, \text{Scaled}_\text{sqerror}_7).
\]
(iii) Then a variable for channel length of 7 is set to 1 or 0 based on whether the current channel length is 7 or not. Then using an exponential averaging, an averaged variable for channel length of 7 is derived.

(iv) After that, it’s decided whether the channel is delayed or non-delayed (e.g. length 4 or 7) by thresholding the above averaged variable.

The angle based demodulator will produce soft bits [SA1] if the switch_4 is enabled. The enabling criteria for switch_4 are given in the table-3.

Table-3, switch_4 enable disable criteria

<table>
<thead>
<tr>
<th>Switch_4 Input status</th>
<th>Switch_4 enable/disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC status: Failed</td>
<td>enable</td>
</tr>
<tr>
<td>Detected channel type: AWGN</td>
<td></td>
</tr>
<tr>
<td>Channel length: 3 or 4 (short)</td>
<td></td>
</tr>
<tr>
<td>Else for any other conditions</td>
<td>Disable</td>
</tr>
</tbody>
</table>

**Soft bit combiner module**

This module takes input soft bit values from four different units- first time demodulated soft bits (S1), re-encoded weighted soft bits (SW2), re-estimated soft bits (SRE2) , and soft bits from angle based demodulator (SA1). Then this unit combines these soft bits to form resultant soft bits.

\[
S_{\text{resultant}} = \{ \text{Weight}_1 * [S1,...,Sn]_{\text{source}_1} , \ldots, \text{Weight}_k * [S1,...,Sn]_{\text{source}_k} \}.
\]

Where, Weight_1,...Weight_k are the different proportional weight factor used for biasing different soft bits generated from four different sources.

\[
\text{Weight}_1 + \ldots + \text{Weight}_k = 1.0
\]

Sometime, some sources might not be present, then the Weight value for it will be 0.

Generally, two algorithms are used:
Averaging weighting method: Here, the resultant soft bits are the simple average of the soft bits received from various sources.

\[ \text{Sresultant} = \{ \text{Weight}_1 \times [S1,\ldots,Sn]_{\text{source}_1}, \ldots, \text{Weight}_k \times [S1,\ldots,Sn]_{\text{source}_k} \}. \]

\[ \text{Weight}_1 = \ldots = \text{Weight}_k = 1 / (\text{Weight}_1 + \ldots + \text{Weight}_k) \]

Proportional weighting method: Here, the soft bits generated from the re-encoding process is given higher weight compared to other sources.

\[ \text{Sresultant} = \{ \text{Weight}_1 \times [S1,\ldots,Sn]_{\text{source}_1}, \ldots, \text{Weight}_k \times [S1,\ldots,Sn]_{\text{source}_k} \}. \]

\[ \text{Weight}_1 + \ldots + \text{Weight}_k = 1.0 \]

\[ \text{Weight}_{\text{SRE2}} > \text{Weight}_{S1} > \text{Weight}_{\text{SW2}} > \text{Weight}_{\text{SA1}}. \]
Figure-8.1, Proposed receiver architecture with adaptive switching

**D) Switching decision for Viterbi decoder (adaptive decoder selection)**

The Viterbi decoder uses trellis structure and computes a metric for each path and makes a decision based on this metric. All paths are followed until two paths converge on one node. Then the path with the higher metric is kept and the one with lower metric is discarded. The paths selected are called the survivors. For an N-bit sequence, the total
number of possible received sequences is $2N$. Of these only $2^{kL}$ are valid. The Viterbi algorithm applies the maximum-likelihood principles to limit the comparison to 2 to the power of $kL$ surviving paths instead of checking all paths. ($L$ is the constraint length, $k$ is the stages). The most common metric used for path error computation is the Hamming distance metric or Euclidean distance metric. Mostly commonly the present day’s receiver uses Euclidean distance based metric, which is more computational intensive.

The demodulator passes the soft bits to the channel decoder for decoding. If in the received signal the presence of noise is less (received signal strength is high), signal fluctuation is less (fading is less) and presence of interference is less (that means in a high SNR and C/I conditions), all the soft bit values from the demodulator are relatively higher. That indicates that the demodulator is very confident about the demodulated bits.

In this condition, using a complex Viterbi decoding structure (computing the Euclidean distance based path metric for several paths) for convolution decoding is not optimal, as, this takes too many cycles which leads to more processing power requirement, more processing time or delay, and increased power consumption. So, in such condition, using a much simpler convolution decoder (which takes only few cycles) is more appropriate to use to get the above mentioned advantages. That will help for improved battery power, faster processing (helpful for faster processing in Multi-RAT scenarios), less processing cycles requirement.

Here, in this work, it proposes two new blocks- (a) soft bit quality indicator (b) simple convolution decoder. As shown in figure 8.2, the ‘soft bit quality indicator’ checks the confidence values of demodulator generated soft bits and indicates whether all the soft bit
values are above a defined threshold or not. If the indication is true, that indicates using
trellis structure the demodulator has given already error corrected bits or removed the
influence of channel noise in the demodulated bits. But those bits are still in the
convolutionally encoded form (as done by transmitter). So, these need to be
convolutionally decoded. Here, a new simple convolution decoder is used as shown in
figure-8.2. This decoder uses the generator polynomials used in the encoder side and
decodes the bits comparing the errors for ‘1’ and ‘0’ bits. This can be considered as a
single stage hamming distance based decoding, which takes very less cycles compared to
Viterbi decoder. Dynamically, the decoder is selected either legacy complex decoder
(Viterbi) or simple one stage decoder based on the soft bit quality indication. If the soft
bit quality is poorer than Viterbi is used else simple decoder is used. During the receiver
operation, most of the time the channel condition is good (especially for some channels
the transmitted power is high) and signal strength is high so, the soft bits are generated
with higher confidence. In those scenarios, convolution decoding without trellis structure
is very optimal with respect processing overhead, time and power consumption. The
existing/implemented Viterbi algorithm can’t be dynamically scaled down to reduced
state or paths so easily, so this proposed method will be very helpful in this regard by
dynamically switching between two decoders, based on trade-off requirement between
error correction gain or processing complexity gain.

The steps followed for the simple convolution decoding will be as below:

(1) The demodulator provides the soft bits with soft bit signs and values. The soft bit
sign indicates the ‘1’ or ‘0’ bits and confidence value indicates about the
confidence about indicating so. The soft bits are input to the ‘soft bit quality
indicator’. That checks whether all the soft bit values are above the threshold or not. If all the soft bit values are above a defined threshold (say Th_Val) then it indicates with a binary output indicating whether soft bit values are Good (1) or Not (2).

As an example, here we have taken GSM SACCH channel encoding-decoding process. There 456 soft bits generated per SACCH block and input to the decoder. Say, if there are 456 number of soft bits are produced for a data block (over 4 bursts) and if all the 456 soft values (absolute value e.g. no sign) are above Th_Val, then this will indicate 1 else 0.

(2) Generally, Th_Val can be determined using empirical method e.g. in the hardware platform check when the signal level is very good and no channel fading is present in such scenarios- what is the minimum value of the soft bits. That can be set as the Threshold value. The same can be done using a system simulator.

(3) Then if indicator value is 1 e.g. all the soft bits are good, then enable simple convolution decoder else enable Legacy complex convolution decoder (Viterbi decoder).

(4) Simple Convolution Decoder (SDC) - takes the input soft bits. Then-

(a) Directly converts the soft bits into hard bits by representation –ve values as ‘1’ and +ve values (including 0) as ‘0’.

(b) Initialize the generator polynomials (G) (same ones are used in the transmitter or encoder side). Generally if ½ rate encoding is used in encoder side then two output bits are generated per input data bit. So, at a time two soft bits will be taken to produce one hard bit on the decoder side. And there will two number
of generator polynomials (G0 and G1). Similarly, it will be set for 1/3 or other rate.

For example, SACCH channel uses 1/2 encoding rate, so two encoded bits are generated per data bits. So, here two consecutive soft bits will be taken at a time to generate a hard bit. Two polynomials used are- \( G0 = 1+D3+D4 \); \( G1 = 1+D+D3+D4 \); Maximum four stages required (D4). Connect these polynomials to generate outputs.

**Software code/logic:**

```plaintext
int G0_0= 1, G0_1=0, G0_2= 0,G0_3=1, G0_4=1; //initialize poly G0 = 1*G0_0 + 0*G0_1 + 0*G0_2 + 1*G0_3 + 1*G0_4
int G1_0= 1, G1_1=1, G1_2= 0,G1_3=1, G1_4=1; //initialize poly G1 = 1*G1_0 + 1*G1_1 + 0*G1_2 + 1*G1_3 + 1*G1_4

Now connect these to generate output-


a[] is array to store the bit values. a[0] represents the first bit and a[4] is the last bit in the bit register. Bits are inserted in a[0] and leaves from a[4].

(c) As it is known that the initial starting state will be all zero state e.g. 0-0-0-0.

(d) Compute G0 and G1 assuming the first transmitted bit as ‘1’ that means a[0] =1 e.g. 1-0-0-0 sequence.

(e) Take first two soft bits S0 and S1 and compare S0 with G0 and S1 with G1 and compute error_0 and error_1 respectively.
(f) If sum of errors e.g. here \((\text{error}_0 + \text{error}_1)\) is less than equal to 1 then considered that assumption is right that means first transmitted bit was 1. Else 0. Like this first hard bit is decoded. Say, here decoded bit (DB) is found as 1.

(g) Then the registered array \(a[4]\) will be filled with these values e.g. now it will be 1-0-0-0.

(h) For the second bit decoding, the register content needs to be shifted right by one bit. As shifting bit is a cycle expensive, so bits can be swapped with value assignment as below software logic.

**Software source code/logic:**

```c
//shift right by 1 bit
a[4] = a[3];
a[3] = a[2];
a[2] = a[1];
a[1] = DB;
```

(i) Then take next two soft bit values and compute the G0 and G1 in the same way assuming \(a[0] = 1\) e.g. then the register sequence will be 1-1-0-0. Check the sum of errors is less than or equal to 1 or not if so, then second it is 1 else 0. Decoded bit (DB) will be \(a[0]\).

Then again populate the register content as earlier e.g. \(a[4] = a[3]; a[3] = a[2]; a[2] = a[1]; a[1] = DB;\)

(j) This process will be continues till last soft bit pairs are decoded. Each time two soft bit are taken so, if there are 456 soft bits are there then total \(456/2 = 228\) numbers this process will continue to generate 228 decoded hard bits.
(k) The same method will be followed for 1/3 rate convolution decoding. There
the number of polynomials will be 3, G0, G1, G2 and at a time three number
of soft bits will be taken to generate one decoded hard bits.

Example software code for this proposed method:

//soft bit quality indicator. Say ‘c_float’ is the demodulated, de-punctured, de-interleaved
soft bits
#define Th_Val 0.200    //or it can be +127 or -127 in fixed point representation
int max_softbits = 456; //max number of soft bits per block
int indication =0;
for(int i=0;i<max_softbits;i++)
{
    if(c_float[i] >= Th_Val) indication++;
}

//simple convolution decoder call
if(indication == max_softbits) //all soft bits are above threshold
{
    int N = 456; //max no of soft bits or encoded bits.
    int a[5]={0}; //array to keep bits as shift register
    //initialize polynomials, as used in the encoder side. Here it is 1/2 rate is used and
    also 4 stages are used. Poly are => G0 = 1+D3+D4; G1 = 1+D+D3+D4;
    int G00= 1, G01=0, G02= 0,G03=1, G04=1;//initialize,
    int G10=1, G11=1, G12=0, G13=1, G14=1;//initialize,
int DB=0; //decoded hard bits
int HB[228]={0}; //output buffer for storing decoded bits

for(int i=0; i<456; i=i+2) //increment I according the 1/2 (or 1/3 rate). Take two
soft bits at a time e.g. i+2
{
    // covert soft bits to 1 or 0 e.g. soft bits to hard bit form
    int sftsign0, sftsign1;
    if(c_float[i]<0)sftsign0 =1; else sftsign0 =0;
    if(c_float[i+1]<0)sftsign1 =1; else sftsign1 =0;
    a[0]=1; //make it zero first to compare decision
    int error0 = (abs( sftsign0 - G0 ) );
    int error1 = (abs(sftsign1 - G1) );
    if((error0 + error1) <= 1) DB =1; //for 1/3 rate – three errors will
    be computed
    else DB =0; //error is greater assuming a[0] bit 1 so, it will be 0.
    //set bits for right shifting operation
    int index = i/2; //based on the rate i/2½ or i/3
    HB[index] = DB; //stored decoded bits
    //printf("decoded hard bit[\%d]=\%d\n",index, DB);
This proposed method will help to optimize processing overhead, time and power consumption without sacrificing FER / BLER.

Figure 8.2. Block diagram of the proposed receiver with this new simple decoder operation in conjunction with the legacy Viterbi decoder [dynamic switching will help to
keep the FER/BLER same as earlier case, and help to save processing & battery power and memory because of the dynamic selection of simple convolution decoder]

Generally in wire-line channel or in wireless channel with good channel conditions, the received soft bit values are higher in confidence. That is case in most of the operating conditions, which is observed during the Mobile’s normal daily operation. So, this simple decoding will be called most of the time and will provide maximal gain.

For the soft bit quality indicator another simpler approach will be to compute the mean and then compare with a threshold that will take further less cycle.

For winning bit generation- the error comparison can be done in another way - assume the first inserted soft bit as 1 and estimate the error_1, also estimate the first soft bit as 0 and estimate the error_0. Then compare the values between error_0 and error_1 and whichever is less just select that bit as winning bit. This will take little bit more cycle but better with respect to performance.
Figure 8.3, algorithm flow of proposed simple convolution decoder

Further optimization for processing power reduction by replacing XOR operation by addition operation:

When computing the polynomials, G0 and G1 above there are some XOR operations involved like-

\[
\]

These XOR (^) operations can be replaced by simple addition operations as below-

Now, do module two division to G0 and G1 to get binary value or divide G0 and G1 to get binary remainder value.

Bit field operations for register: bit filed operator can be used instead of array a[4];

The soft bit quality indication another parameter can be taken into consideration is C/I ratio. In the previous filled ID the description of C/I estimation from the equalizer can be found. If the estimated C/I is above threshold and all soft bits are above threshold then enable this simple decoding method.

36 Important Contributions from this Research work

This dissertation makes a significant contribution to the field of mobile and wireless communications by providing several new as well as improved ways to design the receiver and mobile systems to boost the receiver sensitivity, which is considered to be the most important attribute of the mobile communication receiver.

There are several research papers, books are published and some patents generated from this work as mentioned below.

(a) **Chapter-3**: Various filter structures, types and parameters impact the receiver sensitivity level very much. Here, these impacts are analysed and the optimum parameters design techniques and reference values are suggested to the designers.
Also, an adaptive filter parameters tuning are proposed for achieving optimum performance based on channel type. Two papers are published on these methods and results [47], [41].

(b) Chapter-4: The wireless channel environment is very complex. In this chapter, various wireless propagation environments are analysed and an adaptive detection mechanism is proposed. That helps for best receiver processing algorithm selection based on the detected channel conditions. Several other ideas also found out and proposed for achieving better sensitivity level. One paper is published on these methods and results [38].

(c) Chapter-5: In this chapter, various non-linear approaches are explored and new design techniques are recommended for receiver sensitivity improvement. Also, simple iterative channel estimation technique is recommended. Two papers are published [45], [48]. Two patent applications are filled [34], [35].

(d) Chapter-6: In this chapter new methods for soft-combing techniques are proposed and recommended. These are very useful for sensitivity improvement. Three US patents are filled on these methods [30], [33], [37].

(e) Chapter-7: In this an adaptive combiner unit is introduced to combine these soft bits from first time transmitted soft bits, re-encoded converted soft bits and second time transmitted soft bits using a voting and weighting methods, and then it is passed for second time decoding. This improved adaptive combining process using soft bit diversity provides a significant performance gain for repeated transmission techniques. Three US patents are filled for these methods [31], [32].
(f) Chapter-8: In this conclusion are drawn on different explored and proposed techniques and trade-off analysis. Adaptive receiver architecture is proposed for optimum power-performance trade-off. These are recommended in papers [39], [48]. These contributions will positively help to the wireless telecommunication community, mobile phone design houses and manufacturers to develop high performance and low cost products.

# published 8 US PATENTs from this research work (Please find the detailed information and web link in section-41).

# published 6 Research PAPERs from this research work (Please find the detailed information and web link in section-40).

37 Limitation of this study

Over the last few years, there has been a considerable resurgence in interest in wireless communication. And due to the ever increasing demand for higher data rate, support of more complex applications, seamless hand-over amongst various networks, the wireless system has been evolved over several generations. There are many issues associated with today’s mobile user equipment (UE). Amongst those, one of the important one is sensitivity level, which is the addressed in this research work. But, here, primarily the work was focused on GSM mobile phone sensitivity improvement. Though some of the techniques, recommended here are directly applicable to other mobile radio access technologies like WCDMA (3G) or LTE (4G) mobile user equipment, but there is scope
to further study about the potential design optimization areas for sensitivity improvement for WCDMA and LTE supported mobile phones. Also, apart from sensitivity issue, there are several other issues associated with the mobile devices, these needs to be studied and analyzed for high performance and cost effective mobile phone design.

38 Future Work

In this research work, it is tried to draw the attention of the wireless research communities to improve the receiver sensitivity, as this is one of the most important attributes in a mobile receiver. Though in this work, some of the sensitivity related issues and corresponding design solutions are addressed by exploring and proposing some new design techniques for GSM mobile phones, but still the same exploration needs to be performed for WCDMA and LTE mobile phones. Based on this research work, several other areas can also be explored and analyzed for better RF and Baseband design for mobile phones in other radio access systems. The following topics can be explored further for the future research work [46].

(a) Sensitivity improvements for WCDMA and LTE mobile devices

(b) RF and baseband design optimization for WCDMA and LTE mobile devices for achieving high performance and low cost