1.1.1 PUCCH Receiver

Figure 1 depicts the PUCCH receiver signal chain and the mapping from the signal chain components to the processing resources. PUCCH receiver exists in eNB and RN.

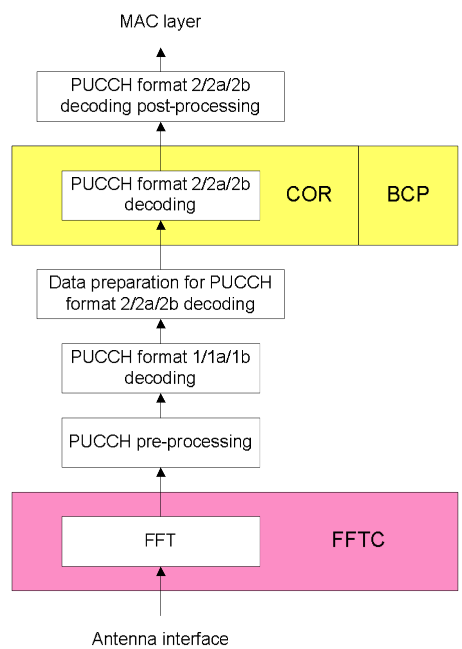

Figure 1: PUCCH receiver signal chain and processing resource mapping

1.1.1.1 Algorithms

The COR submodule uses the pilot-data joint ML decoding for PUCCH format 2/2a/2b. This section describes the algorithm used by BCP. The recommended pre-processing algorithm used by CPU is also provided.

1.1.1.1.1 Algorithm used by COR

The COR submodule does correlation of the received signal over all possible transmitted sequences in an attempt to find the most likely sequence transmitted. The correlation is summed over all antennas as described in the following equation.

$$a_d = \arg\max_{a \in C} \left(\sum_{n=1}^{N_r} \sum_{i=0}^{1} \left| p(i,0,n) + p(i,1,n) + \sum_{j=0}^{N/2-1} y(i,j,n) x'(i,j) \right|^2 \right)$$

where i is the slot index, j is the SC-FDMA data symbol index within the slot, N is the number of SC-FDMA data symbols in a subframe, N_r is the number of receive antennas, p are the received pilot symbols, y are the received data symbols, x' are conjugation of the hypothesized transmitted QPSK symbols, C is the complete set of 2^A possible transmitted sequences, and A is the payload size. The result, a_d, is the most likely transmitted sequence. This equation does a coherent summation within a slot and a non-coherent accumulation over slots and antennas.

COR supports *A* up to 13. The CQI itself only has up to 11-bit payload. But in extended CP mode, when 1-bit or 2-bit HARQ-ACK have to be transmitted, format 2 (instead of format 2a/2b) is used and the 1 or 2 HARQ-ACK bits are concatenated with the CQI bits before Reed-Muller encoding, causing up to 12 or 13 payload bits.

1.1.1.1.2 Recommended PUCCH Pre-processing Algorithm

Upon receiving the UL front-end FFT output for one SC-FDMA symbol on one antenna, pre-processing is performed on the 12 samples of each PUCCH RB. For each [slot *i*, SC-FDMA symbol *j* within the slot, Rx antenna *a*, PUCCH RB *m*], the following two steps are performed. They are similar with the first two steps of DMRS channel estimation.

- Calculate $X_{i,j,a,m}(n) = Y_{i,j,a,m}(n)\bar{r}^{*}_{u,0}(n), 0 \le n < N_{\mathrm{sc}}^{\mathrm{RB}}$, where $Y_{i,j,a,m}(n)$ is the FFT output samples, $\bar{r}_{u,0}(n)$ is the corresponding base sequence [**Error! Reference source not found.**], and $N_{\mathrm{sc}}^{\mathrm{RB}} = 12$ is the number of subcarriers in one RB.
- $2 \times N_{\mathrm{symb}}^{\mathrm{UL}} \times N_r \times M$ times 12-point IDFT's are performed to translate the computing results above from frequency domain to time domain with the formula $x_{i,j,a,m}(k) = \frac{1}{\sqrt{N_{\mathrm{sc}}^{\mathrm{RB}}}} \sum_{n=0}^{N_{\mathrm{sc}}^{\mathrm{RB}}-1} X_{i,j,a,m}(n) e^{2\pi jkn/N_{\mathrm{sc}}^{\mathrm{RB}}}, k = 0, \ldots, N_{\mathrm{sc}}^{\mathrm{RB}} - 1$, where $N_{\mathrm{symb}}^{\mathrm{UL}}$ is the number of symbols per slot, N_r is the number of receive antennas, and *M* is the number of PUCCH RBs. This step can be performed by CPU or FFTC.

For UE *u* in RB *m*, its received signal $y_u(a,i,j)$ for decoder is

$$y_u(a,i,j) = x_{i,j,a,m}((12 - n_{cs}^{i,j,m,u}) \bmod N_{\mathrm{sc}}^{\mathrm{RB}})$$

where $n_{cs}^{i,j,m,u}$ is the cyclic shift of this UE on symbol *j* of slot *i*. In Figure 2, the 12 by 14 grid represents $x_{i,j,a,m}(k); i = 0,1; j = 0,1,\cdots,6; k = 0,\ldots,N_{\mathrm{sc}}^{\mathrm{RB}} - 1$ for antenna *a* and RB *m* in one TTI. The shaded elements represent an example of UE *u*'s received signal $y_u(a,i,j)$ on all 14 SC-FDMA symbols in the TTI. In this example, UE *u*'s cyclic shift for 14 symbols are: 1, 5, 6, 9, 8, 4, 9, 9, 7, 9, 0, 0, 1, 8.

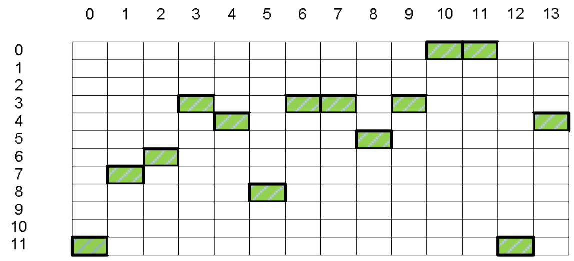

Figure 2: Location of one UE's received signal in IDFT output of one RB (example)

If desired, the samples can be FO compensated before further processing.

1.1.1.2 Data Preparation for PUCCH Format 2/2a/2b Decoding

The time domain signal has to be collected from the 12-point IDFT output buffers and reformatted as described in [**Error! Reference source not found.**] before sent to BCP for format 2/2a/2b processing.

Note that each COR input data value has 16 bits (8-bit I and 8-bit Q) instead of 32 bits.

COR always works on the "5 data symbols plus 2 pilot symbols per slot" assumption.

- In the case of format 2a/2b, the data will be sent to the BCP two (format 2a) or four (format 2b) times for each of the HARQ-ACK bit possibilities. In this case, symbol 5 in a slot is treated as one of the pilots by BCP. Assuming the current hypothesized HARQ-ACK symbol 5 is *X* (table 5.4.2-1 of [**Error! Reference source not found.**]), the received symbol 5 has to be multiplied by *X'* before sent to BCP. That is to say, CPU does correlation multiplication on symbol 5 that BCP doesn't execute.
- In the case of the extended CP (note that formats 2a and 2b are not supported for extended CP as per the standard), two of the four input pilots in a slot are set to zero.

The QPSK mapping defined in the LTE standard is [00→+45°, 01→-45°, 10→+135°, 11→-135°] [**Error! Reference source not found.**], but the QPSK mapping used by COR is [00→+1, 01->-j, 10->+j, 11->-1] for easier hardware implementation. The codeword element (before conjugating) on a pilot symbol is 1 and the corresponding value used by BCP conceptually should be -45°. After conjugating, it is 45°. The multiplication of pilot symbol with codeword element is not performed by BCP, so we have to rotate the pilot data (multiplied by *X'* already for symbol 5 of a slot in the format 2a/2b case) 45° before sending them to BCP.