Uplink SCMA with multiple receive antennas: ideas for TB-ESGA MPA receiver

Bruno Fontan da Silva

January 24, 2020

1 Preliminaries

$\mathcal{A}_q = \{a_1, \ldots, a_q\}$ is an alphabet (set) of q unique complex projections

$\mathcal{C} = \left\{\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(M)}\right\}$ is the mother codebook with M symbols where each symbol is N-dimentional (complex dimensions), i.e. $\mathbf{c}^{(m)} \in \mathbb{C}^N$

the sparse codebook of user j is $\mathcal{X}_j = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\right\}$, where $\mathbf{x}_j^{(m)}$ is the sparse version (K complex dimensions) of $\mathbf{c}^{(m)}$, obtained with the signature $\mathbf{f}_j$ of user j, and each symbol has $K - N$ null dimensions;

$\mathbf{x}_j$ is the transmitted symbol of user j, drawn from $\mathcal{X}_j$ using an uniform distribution such that $P[\mathbf{x}_j = \mathbf{x}^{(m)}] = 1/M$ for $m = 1, \ldots, M$ (*a priori* pmf before decoder sends its estimated LLRs)

$P[\mathbf{x}_j[k] = a_i]$ is the probability that the sparse symbol of user j, $\mathbf{x}_j$, has projection a_m in FN k; it is computed by adding up the probabilities of symbols $\mathbf{x}_j^{(m)}$ where $\mathbf{x}_j^{(m)}[k] = a_i$

$\omega_{j,k,n_r} = \left\{u : |h_{k,u,n_r}|^2 \geq r\, |h_{k,j,n_r}|^2, 0 \leq r \leq 1\right\}$ is the set of strong neighbors of user j in FN k for receive antenna n_r; the set has d strong users, $|\omega_{j,k,n_r}| = d$, being users $u_1, \ldots, u_d$

2 TB-ESGA MPA Algorithm for multiple Rx antennas

Algorithm 1 is a pseudo-code of the implementation of TB-ESGA-MPA with multiple receive antennas.

3 Simulation Scenario 1

Benchmark SCMA $J = 6$ users sharing $K = 4$ resources, using a codebook with $M = 4$ codewords of $N = 2$ complex-dimensions over an alphabet of size $q = 4$.

The base-station has $N_r = 1, 2, 3, 4$ antennas. MPA is used without threshold-approximation ($r = 0$).

LDPC code is short-block length with codeword length $n = 128$ bits, rate 1/2. Each codeword is divided in packets of $k_b = \log_2(M) = 2$ bits, and each packet is mapped to an SCMA codeword. $N_s = 64$ SCMA symbols are sent through the Rayleigh channels $\mathbf{H}_j \in \mathbb{C}^{K \times N_r}$ for $j = 1, \ldots, 6$.

Iterative detection and decoding is performed using $T_o = 3$ outer loops. During the first outer loop, $T_m = 5$ inner MPA iterations are performed. After, when the decoder provides soft-bits to the input of the MPA, only 1 inner MPA iteration is performed in each outer loop.

Algorithm 1 Pseudo-code of TB-ESGA-MPA with N_r receive antennas.

procedure TB-ESGA-MPA-MANY-RX(...)
  for $t = 1, \ldots, T_m$ **do** ▷ MPA iterations
    for $k = 1, \ldots, K$ **do** ▷ For every FN
      for j in ϕ_k **do** ▷ For every VN connected to FN k
        for $i = 1, \ldots, q$ **do** ▷ For every codebook projection of VN j in FN k
          for $n_r = 1, \ldots, N_r$ **do** ▷ For each receive antenna
            Find ω_{j,k,n_r} and $\overline{\omega_{j,k,n_r}}$ with threshold $r\left|h_{k,j,n_r}\right|^2$ ▷ Sets of strong and weak neighbors
            Evaluate $\mu_{j,k}$ and $\sigma^2_{j,k}$ $\forall j \in \overline{\omega_{j,k,n_r}}$ ▷ Gaussian Approx.
            $y' = y_{k,n_r} - h_{k,j,n_r} a_i - \mu_{j,k}$ ▷ Result=Eff.Noise$_{k,n_r}$ + interference$_{k,n_r}$

$$\text{Proj Prob}_{k\to j}(i) = \max_{m_{u_1},\ldots,m_{u_d}}^{\star}\left[\log\left(\frac{1}{2\pi\sqrt{\sigma^2+\sigma^2_{j,k}}}\right) - \frac{\left|y' - \sum_{i=u_1}^{u_d} h_{k,i,n_r}\mathcal{A}_q(m_i)\right|^2}{2\left(\sigma^2+\sigma^2_{j,k}\right)} + \sum_{l=u_1}^{u_d} P[\mathbf{x}_l[k] = a_{m_l}]\right] \quad (1)$$

          end for
          $I_{k\to j}(m) = \text{Proj Prob}_{k\to j}(i)$ for every symbol m where $\mathbf{x}_j^{(m)}[k] = a_i$
        end for
        Normalize messages $I_{k\to j}(m)$ such that $\sum_{m=1}^{M} \exp\left[I_{k\to j}(m)\right] = 1$
      end for
    end for
    for $j = 1, \ldots, J$ **do** ▷ For every VN
      for k in φ_j **do** ▷ For every FN connected to VN j
        if $t < T_m$ **then**
          $\varphi = \varphi_j - k$ ▷ remove previous belief of k to compute the message to k
        else
          $\varphi = \varphi_j$ ▷ Last iteration ($t = T_m$): use messages from all connected FNs to compute final beliefs
        end if
        for $m = 1, \ldots, M$ **do** ▷ For every symbol of the codebook
          $I_{j\to k}(m) = \sum_{k\in\varphi} I_{k\to j}(m) + \log\left(P[\mathbf{x}_j^{(m)}]\right)$
        end for
      end for
      Normalize messages $I_{j\to k}(m)$ such that $\sum_{m=1}^{M} \exp\left[I_{j\to k}(m)\right] = 1$
      if $t = T_m$ **then**
        Use $I_{j\to k}$ as the *a posteriori* pmf of $\mathcal{X}_j$ to calculate LLRs
      end if
    end for
  end for
end procedure

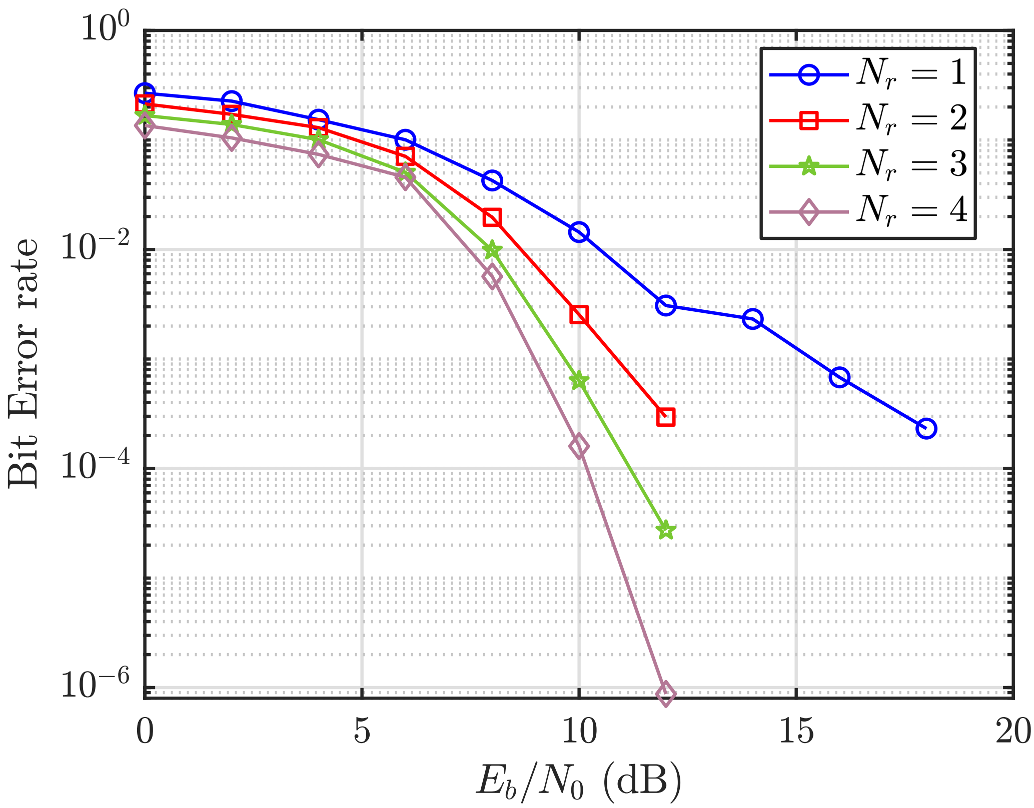

Figure 1: Results for simulation scenario 1.