7.7 Partial Parallel Interference Cancellation

- [Divs 98] introduced a powerful iterative method for non linear MUD. Two features distinguish it from the usual PIC of Section 7.6:
  - use of partial cancellation
  - use of soft nonlinearity for tentative decisions.

  It can also be adapted to the pipeline structure for asynchronous conditions.

- The partial cancellation aspect is similar in effect to "over relaxation" methods. From the summary in [Tan 00], we solve $G\underline{b} = \underline{\xi}$ by

$$\underline{b}^{(j)} = \mu D^{-1}(\underline{\xi} - G\underline{b}^{(j-1)}) + \underline{b}^{(j-1)} \qquad \text{(Jacobi, over relaxed)}$$

where μ is a step size parameter. We recognize this as the steepest descent algorithm analyzed as the mean value of LMS.
  - If $\mu = 1$, then it's Jacobi
  - Ensured convergence if μ is small enough
  - Fastest if $\mu = \frac{2}{\lambda_{max} + \lambda_{min}}$, eivals of G

Relevance here is use of diagonal elements from the previous iteration, as well as off diag (G-S, Jacobi)

- [Divs98] formulate the iteration as a problem of updating <u>statistical</u> estimates of $\underline{b}$ at each stage. They distinguish between a linear mmse estimate $\tilde{\underline{b}}$ and a soft decision $\hat{\underline{b}}$ used for subtractive cancellation. The effect is similar to rewriting the J-OR iteration

$$\underline{b}^{(j)} = \mu D^{-1}(\underline{y} - G\,\underline{b}^{(j-1)}) + \underline{b}^{(j-1)} \qquad G = L + D + U$$

$$= \mu D^{-1}(\underline{y} - (L+U)\underline{b}^{(j-1)} - D\,\underline{b}^{(j-1)}) + \underline{b}^{(j-1)}$$

$$= \mu D^{-1}(\underline{y} - (L+U)\underline{b}^{(j-1)}) + (1-\mu)\,\underline{b}^{(j-1)}$$

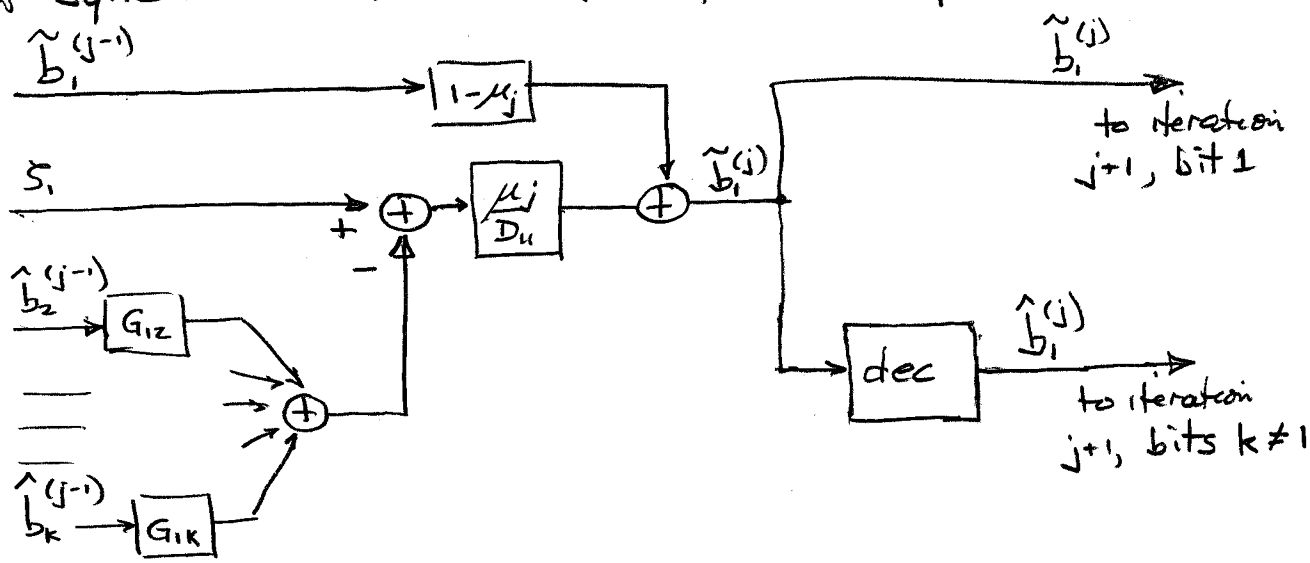
and modifying to

$$\tilde{\underline{b}}^{(j)} = \mu_j D^{-1}(\underline{y} - (L+U)\hat{\underline{b}}^{(j-1)}) + (1-\mu_j)\,\tilde{\underline{b}}^{(j-1)}$$

$$\hat{\underline{b}}^{(j)} = \underline{\mathrm{dec}}(\tilde{\underline{b}}^{(j)})$$

In addition, the dec function corresponding to mmse estimation of b_k, at least for BPSK, is shown to be $\tanh(\tilde{b}_k)$, and for ML estimation, it's $\mathrm{sgn}(\tilde{b}_k)$.

- The receiver structure is relatively straightforward for synchronous. Here it is for bit b_1:



- Does it work?
  AWGN, synchronous, $N_c = 100$, p_1 and p_2 same as μ_1 and μ_2

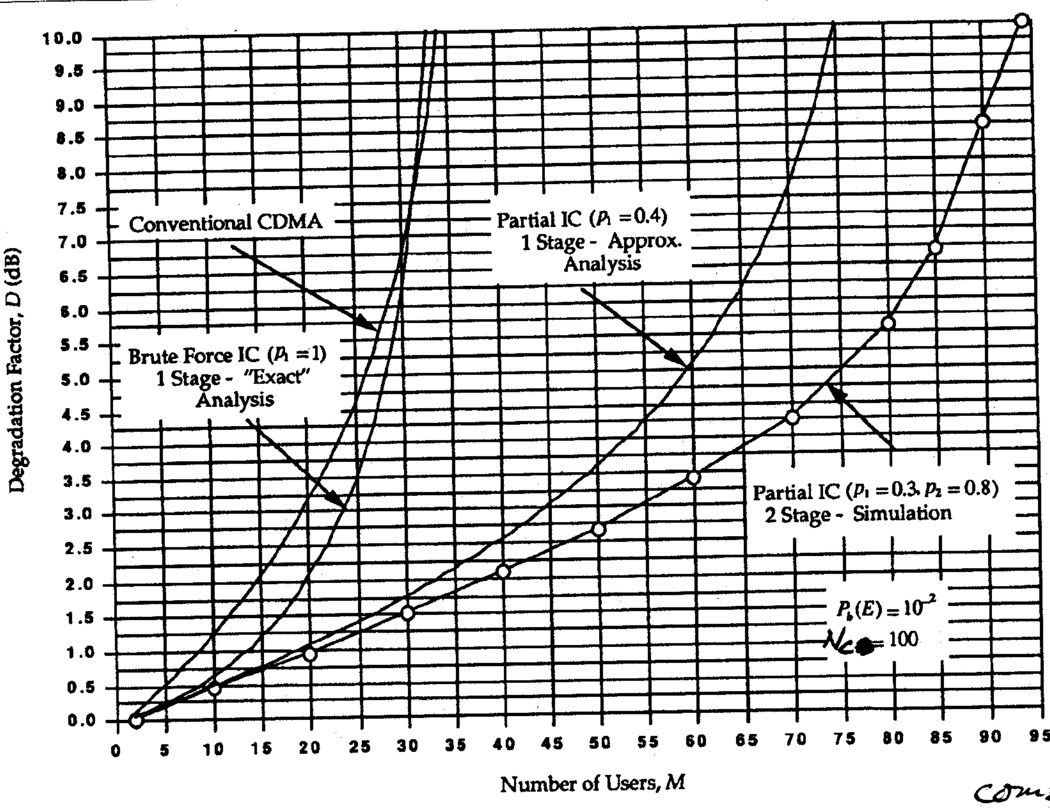


Fig. 6. A comparison of the degradation factors for one- and two-stage linear interference cancellation—equal power users.

linear cancellation

long codes

BPSK

compared to single user

brute force is $\mu_1 = p_1 = 1$ pure Jacobi, sgn.

And for nonlinear cancellation using dec = sgn (not dec = tanh)

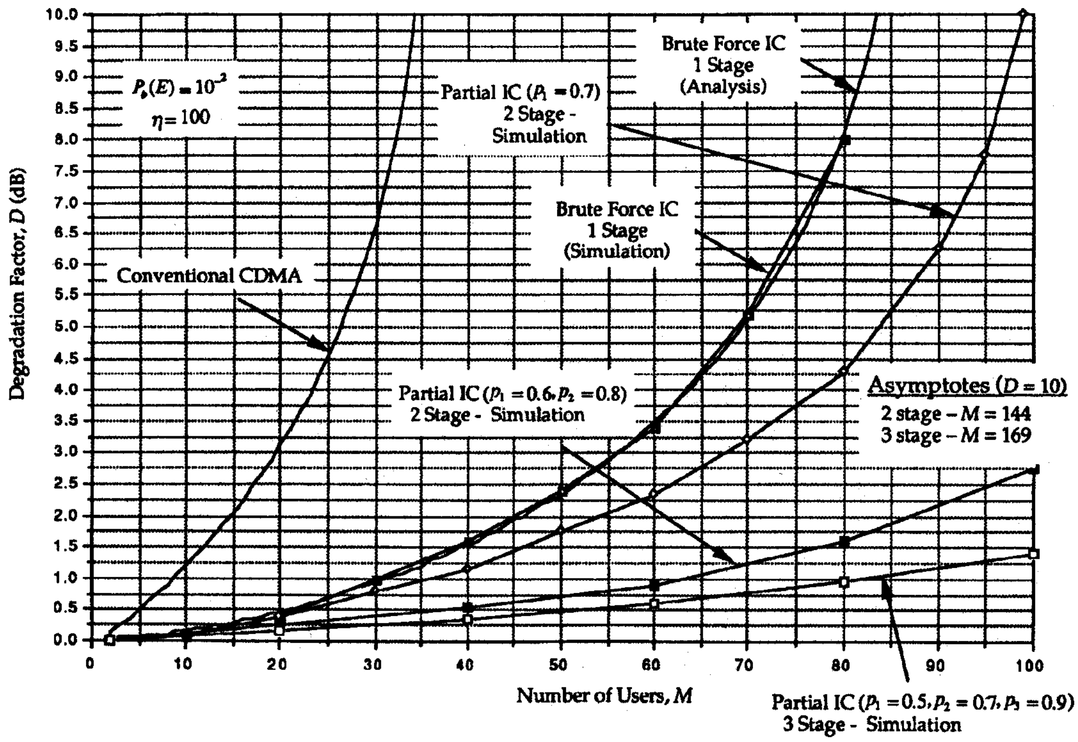


Fig. 7. A comparison of the degradation factors for one-, two-, and three-stage nonlinear interference cancellation—equal power users.

Very impressive – minor degradation, even at high loading ($K = N_c$ and beyond).

Caveat – this is AWGN.

- Readily adapted to asynch with the pipeline (see p 7.3.4).