Further Results on Fast Iterative Coding for Feedback Channels: Multiple-Access and Partial-Feedback

James M. Ooi and Gregory W. Wornell
Massachusetts Institute of Technology, Suite 36-677, 77 Massachusetts Ave., Cambridge, MA 02139, USA
Email: jooi@allegro.mit.edu, gww@allegro.mit.edu

Abstract — The compressed-error-cancellation framework of [1] is extended for multiple-access channels with feedback and single-user channels with partial feedback.

I. INTRODUCTION

In [1], we develop the compressed-error-cancellation framework for coding for feedback channels and apply it to single-user channels with complete feedback. In this paper, we summarize some recent extensions of the framework for multiple-access channels and channels with partial feedback.

II. MULTIPLE-ACCESS CHANNELS

Consider a discrete memoryless multiple-access channel with feedback with channel transition probability function $q_{Y|X_1,X_2}$ and input probability mass functions $q_{X_1}$ and $q_{X_2}$. Let $X_1$, $X_2$, and $Y$ be distributed such that $p_{Y|X_1,X_2}(y|x_1,x_2) = q_{Y|x_1,x_2}(y|x_1,x_2)q_{X_1}(x_1)q_{X_2}(x_2)$. Then consider the following coding scheme based on the compressed-error-cancellation framework:

User 1 (Tx-1):

- Source 1 produces $N_1 = k_1n_1$ message bits to be sent in $k_1$ submessages, each of $n_1$ bits, to Rx.
- Tx-1 sends the first $n_1$-bit submessage using the fixed-length variant of the iterative feedback scheme from [1], taking $\eta^{mb}$ channel inputs, where $n_1/\eta^{mb} = r_1$, where $r_1 < I(Y;X_1)$.
- Tx-1 sends, in order, each of the second through $k_1$ th $n_1$-bit submessages using the same fixed-length scheme.

User 2 (Tx-2):

- Source 2 produces $N_2$ message bits to be sent.
- Tx-2 precedes its $N_2$ bits into the $N_1/n_1$ channel inputs $X_{2}^{N_1}/N_2$; Tx-2 then puts $D_2 < \eta^{mb}$ random filler inputs into the channel while it waits for Tx-1 to finish sending whichever submessage is in progress.
- The channel corrupts the transmitted data according to $p_{Y|X_1,X_2}$.
- Rx feeds the corrupted data $Y_{N_1+D_2}$ back to Tx-2.
- From $Y_{N_1+D_2}$, Tx-2 determines $X_{2}^{N_2}$ (with high probability of success as long as $n_1$ is sufficiently large).
- Tx-2 compresses $X_{2}^{N_2}$ into $N_2/n_1H(X_2|Y,X_1)$ bits, and precedes these bits into $N_1/n_1H(X_2|Y,X_1)/H(X_2)$ channel inputs; again, Tx-2 puts appropriate filler data into the channel.

...