Turbo Codes: Parallel Concatenated Convolutional Codes (PCCCs) with Iterative Decoding

EECS 869: Error Control Coding

Fall 2013

Complete the following tasks. You should submit an e-mail with three .m file attachments (use the e-mail address esp@eecs.ku.edu).

1. **Specify the Trellis Look-up Tables for the Recursive, Systematic, (5,7) Convolutional Code.** In this project, we will be using the recursive, systematic (5,7) convolutional code that was discussed in class. This code will be used twice in the turbo (PCCC) encoder shown in Figure 1 (a) (i.e., it will be used as Encoder 1 and Encoder 2). In the file `RunPcccSimulationTemplate.m`, you will find a "TODO" for specifying the trellis of the DE.

2. **Convert $\mathbf{r}_1$ and $\mathbf{r}_2$ into $\mathbf{P}(\mathbf{c}_1;\mathbf{I})$ and $\mathbf{P}(\mathbf{c}_2;\mathbf{I})$.** In order to do this, you need to specify the values of N_0 and $L_c = 2\sqrt{E_c}/\sigma^2$ in your code. There is a "TODO" in the file `RunPcccSimulationTemplate.m` where you should do this.

3. **Implement the Turbo (PCCC) Decoder.** A block diagram of the turbo (PCCC) decoder is shown in Figure 1 (b). In the e-mail distribution, you have a template decoder `PcccSisoDecoderTemplate.m`. You have already implemented the SISO algorithm in a previous project. Therefore, your task is simply to "wire together" the appropriate modules. Please note that the notation in Figure 1 (b) is different from the notation we have used in class. Here is what you need to know:

   - The upper (code) input to SISO 1 is $\mathbf{P}(\mathbf{c}_1;\mathrm{I})$, and the upper (code) input to SISO 2 is $\mathbf{P}(\mathbf{c}_2;\mathrm{I})$.
   - For the first iteration, the lower (message) input to SISO 1 is initialized to 1/2.
   - Remember that any SISO output that is connected to another SISO's input must be *extrinsic*!
   - The final decision is made using the full output of SISO 2 (not the extrinsic output). There are at least two ways to get this: (1) use the full output before forming the extrinsic output (this is probably the easiest for us), or (2) combine the extrinsic output and the input (this is shown in the figure).

   You do not need to run a full BER simulation; however, the template simulation file is set up to do so. You should verify that your full PCCC decoder works properly. An example BER point is that an information block size of 4096 (code block size of 12288) with 10 decoding iterations has a BER of approx. 2×10^{-3} at $E_b/N_0 = 1.0$ dB.

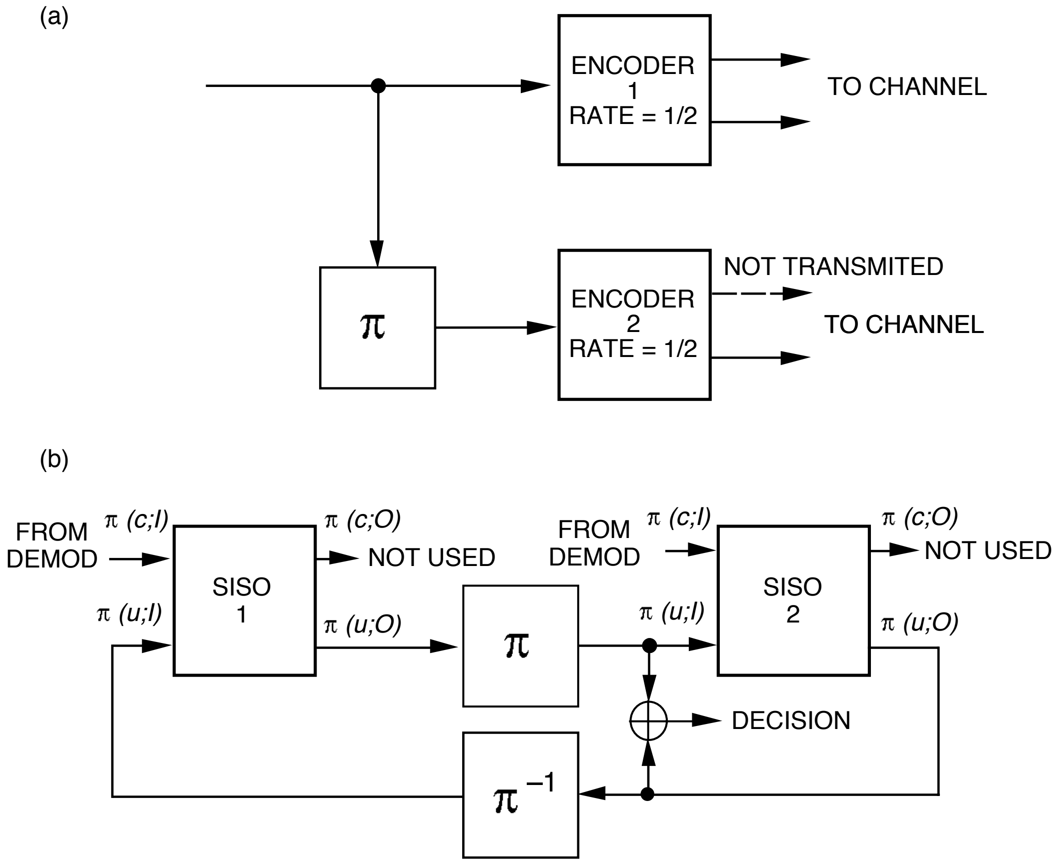

Figure 1: Turbo code (PCCC): (a) encoder, (b) decoder.