If you find this algorithm useful, please cite the following paper. Thanks.

[1] Z. Yang, L. Chen, K. Qin, X. Wang, and H. Zhang, “Perturbation-based decoding schemes for long polar codes,” in Proc. 2025 *IEEE Int. Symp. Inf. Theory (ISIT)*, Ann Arbor, MI, USA, Jun. 2025.

Abstract: For polar codes, the bit-flipping strategy can significantly improve performance of its successive cancellation (SC) decoding. However, the gain derived from SC-flip (SCF) decoding diminishes as the codeword length increases. Addressing this issue, this paper proposes a novel hybrid perturbation-based SC (HPSC) decoding. If the initial SC decoding fails, the algorithm will generate multiple SC decoding attempts, each of which introduces stochastic perturbations to the received symbols. By soft information perturbations, the SC decoding can divert from the initial erroneous estimation and converge to the intended one. Our simulation results show that the proposed HPSC decoding consistently yields stable coding gains over various codeword lengths and rates. With the same number of decoding attempts, the HPSC decoding outperforms the thresholded SCF (TSCF) decoding. Moreover, it can achieve a similar performance as the cyclic redundancy check (CRC) aided SC list (CA-SCL) decoding, without any path sorting and expansion requirements.

Description: This C++ code implements the random perturbation-based SC (RPSC) decoding proposed in our recent work [1]. Note that the same algorithm is also referred to as the perturbation-based SC (PSC) decoding in ITW 2025 [2].

Due to restrictions in the Huawei contract, I can not share any specific parameter settings or the underlying SC decoding algorithm. Readers may use their own SC decoder to replace the algorithm in the program: `**SC(y, msg)**`, where `**y**` denotes the received LLR or the perturbed received LLR, and `**msg**` denotes the SC decoding estimations.

All parameters can be adjusted in `**variables.h**`. The main program is `**main.cpp**`, `**Pol_encode.h**` contains the Polar encoding implementation, and `**decode.h**` implements the RPSC decoding algorithm. Note that the perturbation power is selected according to eq. (16) in [2], since we found that using dynamic perturbation power yields a better FER performance.

These algorithms might be confusing. Please read the paper carefully, and if you still do not understand it, feel free to contact me via email address ZhongjunYang at ieee dot org

Please report any bugs to ZhongjunYang at ieee dot org.

[2] Z. Yang, L. Chen, K. Qin, X. Wang, and H. Zhang, “Improved successive cancellation decoding of polar codes through perturbing *a posteriori* LLRs,” in Proc. 2025 *IEEE Inf. Theory Workshop (ITW)*, Sydney, Australia, Sep. 2025.