/snr_ser:

/direct：存放了不同情况下通过 ruo_main.m 跑出来的 snr 和 ser。

/direct/11.16：存放了经过信道发送数据补正后的 snr-ser，（见/vol_save/11.12_test）。

/direct/11.23：存放了在更正传输错误前后的 snr-ser

/direct/11.25：存放了用来 debug 的数据，见 ruo_debug.m。

/direct/12.1：将两次接收信号（补正信号和被补正信号）重新同步后，跑出来的 snr-ser。

/direct/12.2：在发送信号幅度很大时 ser 会上升，因此储存下来一些数据用来看 ser 上升的原因，对应程序 ruo_debug。

/direct/12.3：在发送信号前加了一段幅度很大的导频用来定位同步点，并且将两次接收信号（补正信号和被补正信号）重新同步，跑出来的 snr-ser。

/vol_save:

/11.4: 每个/amp 下有三个文件：errnum_save.txt、signal_ori_save.mat、signal_received_save.mat，两个.mat 文件中存放了均衡错误数量超过 50 的 400 组 signal_ori 发送数据和 150M 采样率的 signal_received，.txt 中存放了这四百组的均衡错误数量。对应程序：ruo_main_vol.m。

/11.9_test：每个/amp/mat_location 下存放了在该 amp 下，取/11.4 文件中对应/amp 中的 signal_ori_save.mat，取对应的 signal_ori_save_mat_location 作为发送信号，每个发送五次，并记录 signal_demod_ls、signal_downsample、signal_fin

和 signal_ori_save_mat_location 存放在 signal_demod_save_amp40_loc10.mat、signal_downsample_save_amp40_loc10.mat、signal_fin_save_amp40_loc10.mat、signal_save_ori_amp40_loc10.mat 中，errlocation_save_amp40_loc10.mat 里面存放了这五次的均衡错误位置，err_number_amp40_loc10.txt 里面存放了这五次的均衡错误数量，对应程序: ruo_main_voltest.m。

/11.4_test_2：每个/amp 下有一个.txt 文件，存放了以/11.4 中对应/amp 中的 signal_received_save.mat 里的数据作为程序中的 signal_received，进行速率转换、同步、均衡，再与/11.4 中的 signal_ori_save.mat 进行比对，看看错误数量和/11.4 中是不是大致接近，对应程序: ruo_main_voltest2.m。

11.11_test：1 路信道为实验组，4 路为对照组，两路信道发送同样的数据。尝试通过对比实验组和对照组，来删除通过 1 路信道的 160M 信号中出错的点。其中，1 路信道发送两次同样的数据，分为实验组和实验组 1。如果对照组无法删除实验组的全部错误点，那么尝试通过实验组 1 来对实验组进行补正。

unquit_num_amp40_loc10.txt 中存放的是对照组的长度、两个实验组的长度以及两个实验组中未能删除点的个数，以及两个实验组重合的错误点的个数，最后一行是 20 次 exp_time 中有多少次两个实验组有重合点。corrindex_save_amp40_loc10.mat 中存放的是两个实验组重合点的坐标。errloc_save_amp40_loc10.mat 中存放的是实验组中未删除点在 160M 数据中的位置。errloc_save1_amp40_loc10.mat 中存放的是实验组 1 中未删除点在 160M 数据中的位置。对应程序：ruo_test.m。

11.12_test、2、5：与 11.11_test 相同，三个文件夹对应的发送数据长度不同。txt 文件里存放了发送错误的数量以及未能补正的数量，以及实验组 1 和 2 相关错

误点的数量。

ruo_calculate_ser.m：集合了计算 snr、均衡、计算错误点个数的功能。

ruo_channel_coefficient.m：用于仿真生成信道参数。

ruo_debug.m：用于往 mat 里存用来 debug 的数据。

ruo_debug2.m：用于从 mat 里读数据，并用这些数据 debug。

ruo_debug3.m：现在这个 m 文件里写程序，没问题再放进 main2.m 中。

ruo_filter_gen.m：如果 origin_rate 为不规则的速率（如 1.17e6），那么在速率转换时会因为公倍数过大而占满计算机内存。为了解决此问题，编写了 ruo_filter_gen 程序。在该程序中，假如 origin_rate 设为 1.23456e6，那么会通过计算，将该速率转换为相差不超过一定范围的新的 origin_rate，新的速率的公倍数不会很大，规避占满内存。

ruo_load_data.m：用来从 mat 里加载数据，用在 ruo_plot.m 里。

ruo_main.m：主程序，用于在平台上发送数据、接收数据、速率转换、更正传输错误、同步、均衡。

ruo_main2.m：ruo_main.m 的备份，和 ruo_main.m 功能一样。

ruo_main_vol.m：用于生成/vol_save/11.4 中的数据。

ruo_main_voltest：用于生成/vol_save/11.9_test 中的数据。

ruo_main_voltest2：用于生成/vol_save/11.4_test2 中的数据。

ruo_pam4_send.m：用于发送 4pam 数据。

ruo_pam4_send_correct.m：用于在 1 路信道发送补正数据，即 11.11_test 中的实

验组 1。

ruo_pam4_testsend.m：用于 ruo_test.m。

ruo_pam4_testsend2.m：用于 ruo_test.m。

ruo_pam4_volsend.m：用于发送从.mat 中导出的数据，用在 ruo_main_voltest.m 里。

ruo_pilot_gen.m：用于生成导频。

ruo_sam_rate_con.m：用于速率转换。

ruo_signal_equal.m：用于均衡。

ruo_signal_equal_ls.m：用于进行 ls 均衡,和 ruo_signal_equal.m 中不同的是，equal.m 需要的参数是粗同步点，equal_ls.m 需要的参数是精同步点。

ruo_signal_syn.m：用于同步。

ruo_signal_syn_recorrect.m：信道 1 两次发送的信号进行精同步的时候会有一两个相位点的差距，用这个程序来修正这个差距，让这两个信号的图像可以完全重叠。

ruo_test.m：用于生成/vol_save/11.12_test 中的数据。