NACIT策略总控介绍

# 前言：

如果要笔者写策略总控的介绍说明书，那么就必须先讲一下NACIT团队的一段恩怨情仇故事～啪！～（惊堂木）

话说天下大势，分久必合，合久必分。天下江山少安定，江湖儿女多情仇。

仰观天近星辰大，俯视地广世界清。苍穹茫茫，原野苍苍，各方强者雄踞一方，敢问世间何人为王？华夏陕北之地，物华天宝人杰地灵，中原至强之者皆屯与此，其中有名QDH者，振臂一呼，群雄始成联盟，后世名为NACIT。

又有一人，陕北延安人，姓周名二喜，身长七尺，面色黝黑，其貌不扬，其才甚伟，师从陕北贤者刘。二喜幼时即慕古时墨家机甲控制之技，故甚精于控制之术，QDH稀其才，故揽之。后世有图为证：



时值NACIT七人众行往帝都签证之际，二喜未曾同行，故后独往之。

是日天公不美，农者收成欠佳，可谓仓也空井也空，二喜暗惊道：此真乃大凶之兆！签证之时，二喜视签证官，此女全身黝黑，铜铃大眼。二喜惊其黑，孰视之，又弗如远甚，心自大惊。呈其资料，黑女观之，怒斥之：What are you 弄啥嘞？掷白条于二喜资料之上，遂悻悻而去。

二喜归长安，众人皆叹之，并助其再往帝都签证。孰知二次犹未竟。三次再去之时，余七人即将腾云而去，二喜知其难同其行，叹曰：天意如此！

七人行前，二喜仍使尽全身解数，极尽控制之道，同七人披荆斩棘。得二喜助力，七人才得以顺利而去。

七人众中，有河南汴京人氏，人称UPC自动化第二浪，面如冠玉，貌若潘安，身长八尺，容貌甚伟，琴棋书画医卜星象样样皆通，恍若北宋东海桃花岛黄氏药师再生。其人于NACIT中精于识别之道，尤敏于颜色，故七人众中被冠为首席鉴黄师之名。二喜既困于陕北，便将其控制之道授于浪者。浪者深感其德。

故浪者亦通控制之道。

欲知后事如何，请看浪者空间檄文数篇：The Travel of San Diego……

啪～（惊堂木）都散了都散了……

小说时间结束。综上所述，对笔者来说，策略总控和视觉系统的搭建并不一样，因为策略总控并不是笔者构建的框架，笔者也只是受人之托在赛场进行维护而已。另外由于各人有各人的编程风格，在这方面笔者与二喜兄常有摩擦，时至今日也不习惯他的编程风格，所以文中不免有与二喜兄编程风格的对比之处。特此声明，对比本身毫无半点嘲讽之意，只是个人主观想法，各位观众老爷可以自己对比取舍……

笔者比较喜欢把各个功能拆解开，并使其模块化，二喜兄的思路基本上是把所有功能都直接放在了一个箱子里。在笔者个人习惯之下，维护代码的工作还是可以做到的，但讲道理，现在总结二喜兄总控的结构框架，确实有一点不知道从哪里下手，或者无法总结的感觉（依旧没有任何嘲讽的意思）…… 另外，该总控程序的重点在于其运作原理，感觉并不用每句都看懂，所以各位观众老爷只需要浏览代码，了解其工作流程即可。

所以在策略总控的讲解部分，肯定不如视觉系统的讲解来的详细，望各位观众老爷手下留情高抬贵手，别给我一个负分让我滚粗就好了……

# 策略总控大致介绍：

策略总控在笔者看来总共分两大部分，一是变量定义部分，二是程序总控部分。

1. 变量定义部分：

主要包含了控制精度、环境物理参数、各任务过程中需要的变量、与下位机通讯的协议等；

1. 程序总控部分：

程序总控主要是一个主要包含了总控程序的所有函数。包括：

1. 数据通讯函数；
2. 任务控制函数（重要）；
3. 图像处理函数；
4. 其他函数；

# 变量定义部分：

变量定义放在三个头文件中。

1. global.h:

主要存放一些参数，包括一些物理参数（如大气压强），上位机控制的精度范围，任务阶段的枚举变量等；

注：

1. 私以为该头文件包含的变量有点杂乱，所以这里不多赘述；
2. 关于任务目标阶段的枚举，是从笔者开始维护代码之后加进去的。所以在看任务控制部分时，会有两种控制风格：全篇if…else…的是二喜兄的控制代码，用switch切换各个阶段的控制代码是笔者所写。
3. mission\_param.h:

主要存放各个任务中的控制参数。后来又实时更新并保存了赛场环境的实时数据，存储并作用于程序中；

注：平常笔者维护代码绝对有随时加标注的习惯，如果看到有大篇幅代码没有备注的情况，那一定是那段每天睡眠不超过两小时的时间里没有写注释的时间与精力了……

1. thruster.h:

主要存放传给下位机的控制协议，以及判断各传感器是否达到控制精度的标志位；

# 程序总控部分：

本质工作流程在于以500ms为周期的循环更新。实现代码如下：

QTimer \*time\_clock = new QTimer(this);

connect(time\_clock,SIGNAL(timeout()),this,SLOT(AuvMainControl()));

time\_clock->start(CONTROLTIME);

该段代码使用Qt的信号槽机制，使用定时器功能，每一个CONTROLTIME之后，便执行一次AuvMainControl槽函数。此处CONTROLTIME = 500ms。

相关代码全部在auvautocontrol.h, auvautocontrol.cpp中。

注：理论上执行一次控制指令的周期应该是500ms，但实际上运行时我们发现实际的周期经常远大于500ms。笔者认为此现象发生的原因应该是槽函数AuvMainControl自身运算会占据很长的时间，比如图像处理消耗巨大的系统资源，必然会延长函数的运行时间，所以真正的运行周期应该是500ms + 槽函数自身运行时间。

这一点解决方案，笔者会在闲暇时间再考虑，望各位观众老爷也在这方面上多做思考。

其他函数主要还包括：

1. 主控程序初始化函数：

即AuvAutoControl()的构造函数。它连接了信号与槽函数，并将很多数据都进行了初始化。

1. 数据通讯函数：

主要包含与串口通信的操作，包括打开对应串口、发送串口信息等操作函数。包括：

1. 打开两个水声模块串口函数OpenComUnderAc1(),OpenComUnderAc2()
2. 打开下位机串口函数OpenComExecute()
3. 下位机串口发送数据函数on\_ExecuteSend(), ComSend2Execute()
4. 读取方向水声模块函数ReadFromUnderAc1()
5. 读取深度水声模块函数ReadFromUnderAc2()
6. 读取下位机串口数据函数ReadFromExecute()
7. 向下位机发送角度、深度指令Send\_Angle(), Send\_Depth()
8. 任务控制函数（重要）：
9. 任务切换函数SwitchTask(): 功能在于各任务之间的切换，具体实践过程没啥讲的，请自行看源代码体会……
10. 任务控制函数TaskControl(): 功能在于控制当前任务，说白了就是个选择器，具体依旧没啥讲的，请自行看源码体会……
11. 各任务函数：

以M开头的系列函数，用于控制对应的任务；

注：

1. 开始任务M\_Start(), 过门任务M\_ThroughDoor(), 风格过门任务M\_Navigate\_channel()为二喜兄所写，笔者维护；撞球任务M\_Scuttle\_ship(), 投放标志物任务M\_Weight\_anchor(), 水声定位任务M\_Process\_UnderAC(), 八边形浮起任务M\_Bury\_treasure(), 任意水声定位任务M\_RandomPinger()皆为笔者所写；
2. 投放标志物任务、水声定位任务、八边形浮起任务、任意水声定位任务没有足够时间调试，所以表现不佳；其中投放标志物任务思路足够缜密，虽表现不佳，但依旧认为可以参考；而后三个任务完全是为了当时冲分而写，所以参考价值较小……
3. 图像处理函数：

直接调用视觉类对象Vision中的函数，即可完成图像处理部分的任务。工作流程如下：

1. 初始化视觉系统：该步骤在主控程序构造函数AuvAutoControl()中完成；
2. 捕获视频：此部分包含两种模式：调试模式与实测模式。调试模式下使用视觉系统中从视频文件获取图像帧序列，实测模式下使用Guppy Pro相机获取实时图像帧序列；
3. 图像预处理；
4. 按照不同的任务，顺序运行对应任务的二值化处理函数与分析图像函数；
5. 其他函数：
6. 记录函数OpenFile()：

总控程序运行时，用于记录航行器每个指令周期的各参数状况、图像数据状况、向下位机发送的指令。

与之对应的CloseFile()仅用于关闭文件流，不多赘述。

注：笔者认为，该部分虽然有很多漏洞，很大程度上有待改善，但该部分对调试至关重要，由于它的存在调试中发现了很多问题。

1. 格式转换函数StringToDec(), StringToDec2()
2. 安保函数Safety()，防止航行器潜入深度过深；
3. 各种运行界面按钮函数，不再赘述；

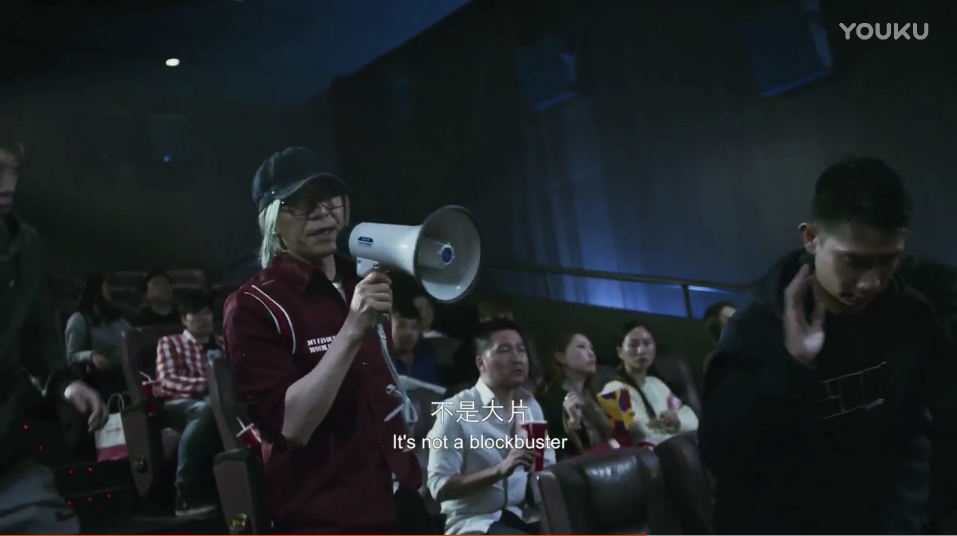
后记：

总控部分，感觉仍有很多地方待改进，这段时间笔者也将在闲暇之余考虑如何在Linux系统下将下一代的软件平台与软件框架搭起来，软件平台与软件框架搭建起来之时，也即是老朽退居田园品茶抚琴之日。请各位观众老爷静候的同时，也好好研读笔者写的说明书与源码，并在此段时间内自觉在网上找一些小项目作为练手，提高自己的实践能力。

下一次看到凌晨四点圣地亚哥的已经不是我们，而是你们这些年轻人。

NACIT装逼顾问琦少 书于丁酉年壬寅月丁亥日

2017.03.01





**这次没有彩蛋……**