先说一下我们机器人的相关特性。

我们组的机器人特别厚实。在所有参赛队伍中，我们的机器人重量算得上是中等偏上的吧。但是我们机器人的底盘特别小（为了适应参加对抗赛的小型尺寸），且机械臂的重量也比较大，因此，底部运动便没有那么灵活。

总的来说，主要有这几个难点：重心不稳，轮子打滑；行动笨重，控制延迟；巡线不准，控制出错。

前期，因为机械臂向外伸展时，会导致整个机器人的重心前倾，导致后边的轮子抓地不牢，便出现轮子打滑现象。轻则举步维艰，重则转圈乱跑。后面，通过在机械臂后方安装两公斤的配重来解决了这个问题。

前期，我们的底座使用的是25直流电机和2HL-70全向轮，电机输出能力较小，全向轮抓地性也不够好。这样造成可控速度范围很小。用pwm波控制直流电机速度时，占空比太小，则在高负荷下根本无法转动。这样其线性控制度也特别差。另外，各个电机和轮子的运行状况也不尽相同。同样的pwm控制值，有些轮子可以转动，有些还不能转。有些轮子已经可以转得飞快，有些轮子一顿一顿的。这就造成控制时要么很快就跑偏了，想调回来的时候却又转不动。

后面将电机换成了37直流电机和中鸣全向轮，电机的输出能力增强，全向轮的抓地性也好了很多。现在，才勉强可以走稳了。接下来，开始解决巡线时的大问题。

我们对巡线的精度要求特别高，因为机械爪抓取物品的位置是固定的，机械爪张开后的宽度不比物品宽多少，这就是底座巡线精度与机械臂的转动精度的最大容错值。我们的巡线一共分为四个步骤。分别是出发前的精确调整，冲出十字路口，巡线至目标位置，精确调整。

控制直流电机，不像步进那样强有力的令行禁止，尤其是背着这么重的一座小山。让它动，它的速度会有一个积累过程；让它停，它的惯性让它划过边界。更可恶的事，它的特性具有一定的概率性。以到达十字路口为例。检测到十字路口，则速度马上将为0，可它有时会冲出十字路口小半格，有时竟然可以达到大半格。这时，调整到十字路口正中心，则十分困难。如果一小步一小步的往回挪，则需要特别长的时间，如果再次用巡线的方式持续给信号，则又可能再次冲过十字路口。为了去除这种过冲效应，于是采用反速刹车机制。

反速刹车就是为了在巡线时快速停下。当巡线模块一检测到十字路口，则马上将速度反向。在巡线时，因为功率经过长时间累积，速度已经达到一个较大的稳定值，所以需要采用最大速度作为反向速度，其制动时间才可以相应减少，增加反应灵敏度。这样做之后，过冲效应确实减小了，不过刹车也可能刹过头。为了后面的进一步调整，我们必须调整刹车参数使得刹车决不能过头。

在各个控制之间，延时也是必不可少的。之前以为刹车已经能做到令行禁止了，刹车结束就马上检测当前巡线状态，然后马上再给出控制信号。然而每一个信号都是对之后有影响的，刹车这一本着消除这因影响的举措也有这影响。向后刹车后，检测巡线信号，发现过冲了，其实那信号也许不是从十字路口冲出去的信号，而是刹车措施将机器人拽回来的信号，然后马上再给他向后拽的力，于是乎，又被拽到了十字路口的另一边，这时，就难以判断了。

无数次的测试经验告诉我，直流电机绝不是冲击响应！你如果给它一个脉冲信号来控制它的基础动作，那么当两个脉冲信号隔得很近的时候，控制便会发生干涉，速度便会累加，便会脱离控制。

所以除了到达目标位置加了刹车动作，巡线时的微调也加了刹车动作，只不过这个动作更小，控制参数也需要根据实际重新更改。力求每一个控制函数都是独立的。如果巡线的微调不加刹车动作，则整个巡线过程便有些拖拉机飙车的感觉，明明不快，却给人一种摇摆不定的感觉。

我们的巡线思路类似于增量调制，前进速度与调整速度是两个非常重要的参数。

所以这里面的许多控制参数都需要根据实际多次试验来确定的。

此后略去千言万语，一切玄机尽在代码中。