# 【开源+讨论】底盘功率限制方案+超级电容

## 当前使用方案介绍

为了让更多新加入的队伍尽快解决底盘功率限制的问题，在此开源一下我们用了两年的功率限制方案。**如果有更好的方案或者改进的方案，希望可以写帖讨论。**

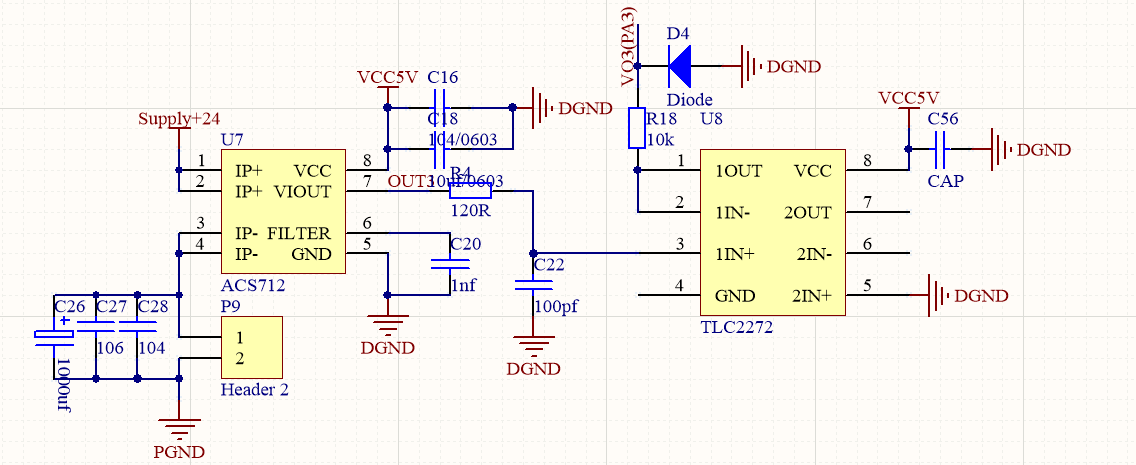
### 底盘功率限制

在正常的使用下，电池电压变化不大，所以底盘功率限制更多体现在对**裁判系统底盘输出电流**的限制。由于我们的控制信号是给每个底盘电机发送的，所以要控制裁判系统底盘输出电流，就需要**对每个电机的电流进行控制**。

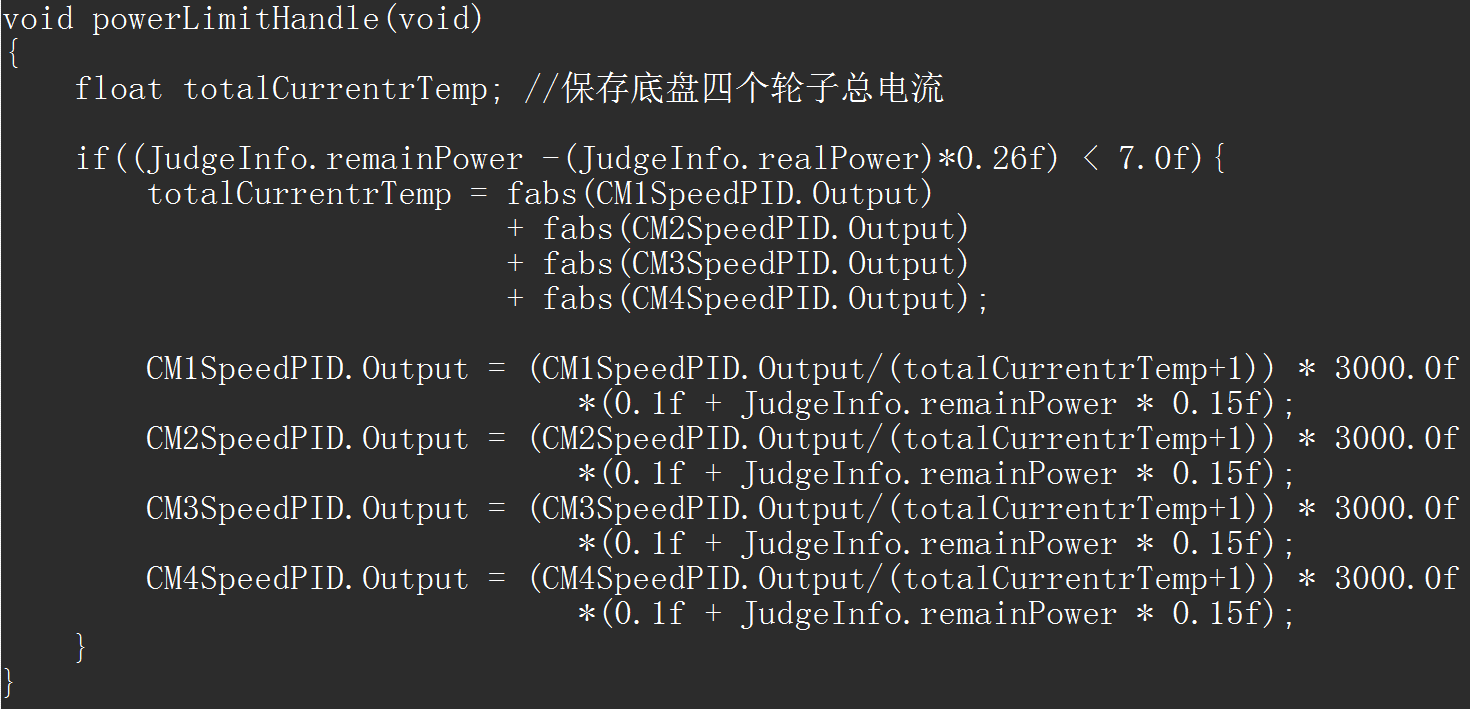
既然要控制每个电机的电流，就需要知道当前电机的母线电流，有两种方法：

1. 使用C620电调反馈的转矩电流，以及电机手册上的负载特性曲线和特征参数，根据公式可以计算出电机母线电流。（*P.S. 道听途说的，我没试过，有兴趣的可以试一下*）
2. 自己做一个**电流检测模块**，对输入各个电机电流进行检测，然后再反馈给主控。

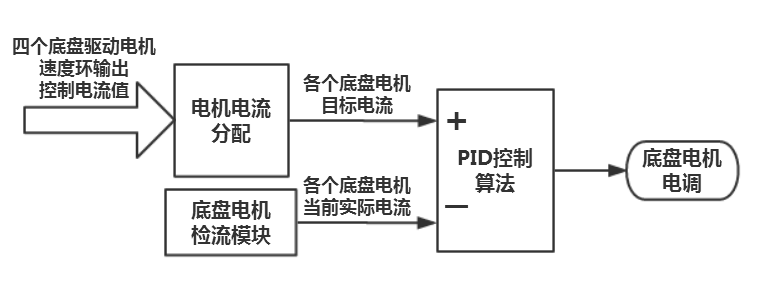
我们使用的方案是自己做一个电流检测模块。我们使用霍尔检流（芯片：ACS712）方案，然后使用**STM32F103C8T6单片机**读取数据，再通过**MAX3051芯片**以CAN总线的方式发送到主控。



同时**读取裁判系统数据**，根据反馈的剩余缓冲能量和实时功率数据，对输出的目标电机电流进行分配，分配程序如下图。（*P.S. 不要问我这个分配的原理，我也不清楚，我原本设计的不是这样的，后面交给别人调程序的时候，就变成这样了，因为能用就没理过了C:\Users\ADMINI~1\AppData\Local\Temp\Z}4JA~3{S79KP7UYELJH){7.gif*）



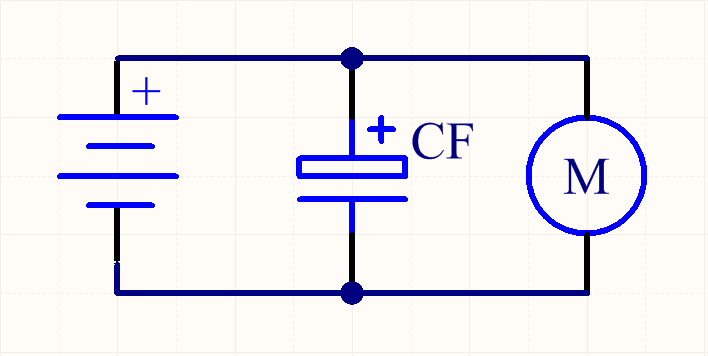
进行电流分配后，将各个电机速度环计算输出的电流值经过电流分配调整后作为每个电机电流环的目标值，再使用电机检流反馈的当前各个电机电流作为每个电机电流环的反馈，进行PID控制算法运算，再把计算结果发送到各个电机的电调。



剩余的就是XJB调参大法了

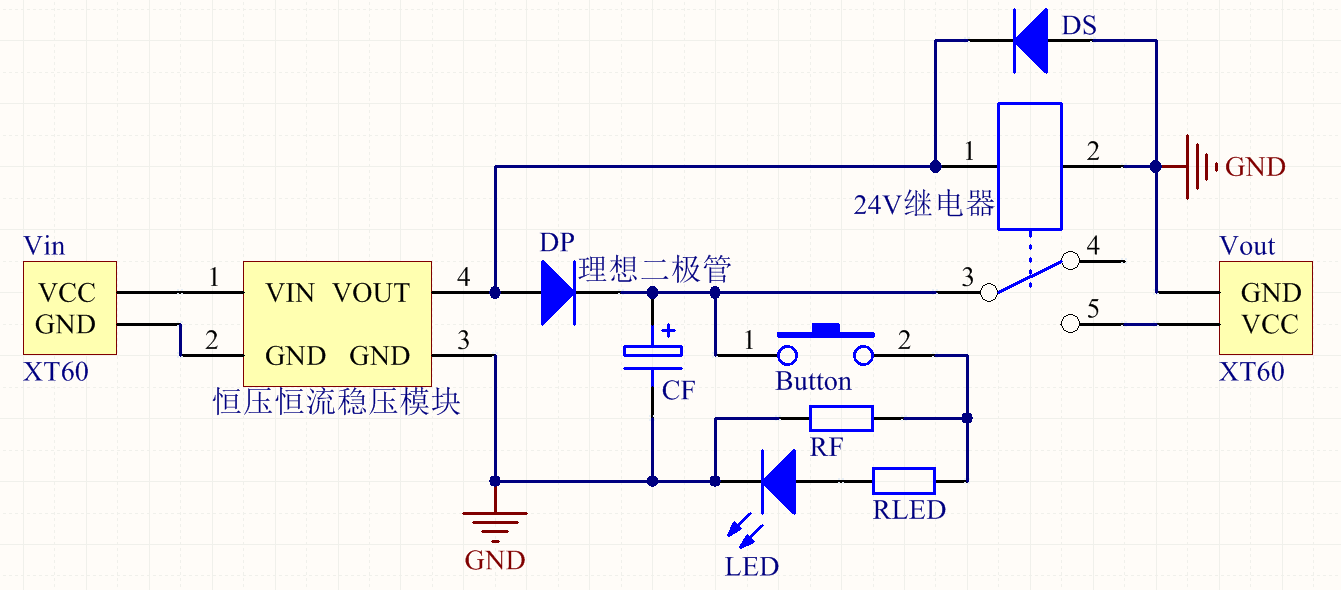
### 超级电容

经过上述的功率限制后，基本上机器人会出现启动加速慢、制动距离远和爬坡无力等现象，这是加一个超级电容就可以改善这种情况。超级电容最直接的方法就是如下图：

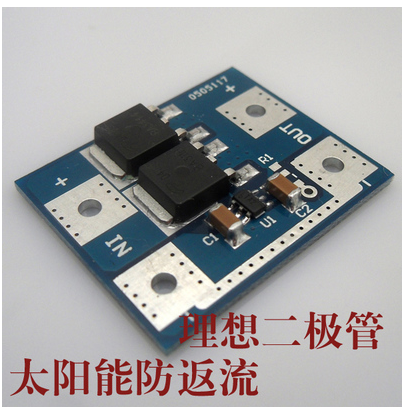


然而这种方式是会出现僵尸车的现象，导致你过不了检录。并且如果电容没电状态，这时给电容充电，电容相当于短路，这时充电电流非常大，会出现上电超功率死掉的情况，所以最好加一个限流的电源。

所以对上述电路进行修改，加入**恒流恒压稳压模块**，**理想二极管**，**继电器**，**手动放电电路**。设置恒压恒流稳压模块的输出电流为功率限制的最大电流，比如步兵为3A。



其中，步兵用的超级电容模组参数为10F/32V。使用的恒压恒流稳压模块，理想二极管如下图：



**注意，网上买到的恒流恒压模块的大多数是低边检流，而裁判系统用的是高边检流，所以恒流恒压模块输出后级电路的地只能接到恒流恒压模块输出的地，不要和其他电路的地接在一起，不然恒流恒压模块无法起到限流的作用。**详情自己查找高边检流和低边检流的特点。

模块链接：

电源：https://detail.tmall.com/item.htm?id=15808996391&spm=a1z09.2.0.0.67002e8dUq0Bft&\_u=42la61k661eb

理想二极管：<https://item.taobao.com/item.htm?spm=a230r.1.14.38.17ec6f23Buj2xH&id=534140581745&ns=1&abbucket=20#detail>

电容：

https://item.taobao.com/item.htm?spm=a1z09.2.0.0.67002e8dUq0Bft&id=561262957496&\_u=42la61k6e9b3

## 想法讨论

在编写这个教程时，笔者发现整个方案其实还存在许多缺陷，同时也有一些想法，然而由于时间关系没法一一测试比较（*P.S. 其实更多是因为比较懒*），在此笔者叙述一下想法，希望可以有人开贴讨论一下具体的方案设计或者提出更好的想法。

### 电机电流分配

接着是底盘电机电流分配逻辑。这里我说一下我之前的想法（以步兵为例）。

已知裁判系统对功率检测是100ms一次，考虑需要有控制余量，根据当前反馈的剩余缓冲能量和实时功率数据，预测一下200ms后的剩余能量，如果小于7J则认为进入即将超功率状态，需要进行电流分配。

if((GameInfo.remainPower - GameInfo.realPower\*0.2f) < 7.0f) {

根据当前的电压，计算满功率(此处是80W)情况下的电流值CurrentFullPower (单位为mA)，以及各个底盘驱动电机的速度环输出目标控制电流值的绝对值之和CurrentTargetSum。

CurrentFullPower = 80\*1000/ GameInfo.chassisVolt;

CurrentTargetSum = abs(CM1SpeedPID.Output) + abs(CM2SpeedPID.Output) + abs(CM3SpeedPID.Output) + abs(CM4SpeedPID.Output);

根据各个底盘驱动电机速度环输出目标控制电流值对满功率电流CurrentFullPower按比例进行分配。同时裁判系统允许使用缓冲能量，假设在200ms时使用完缓存能量根据图4-1裁判系统对底盘功率检测流程图，得

其中，,,分别为0ms、100ms、200ms时剩余能量，为实时功率，为限制功率，

等号两边除以电压

规则限制功率为80W，即，所以得到以下代码，对底盘驱动电机速度环输出电流值进行调整。

CM1SpeedPID.Output = (CM1SpeedPID.Output/CurrentTargetSum) \* CurrentFullPower \*(1 + GameInfo.remainPower \* 0.0625f);

CM2SpeedPID.Output = (CM2SpeedPID.Output/CurrentTargetSum) \* CurrentFullPower \*(1 + GameInfo.remainPower \* 0.0625f);

CM3SpeedPID.Output = (CM3SpeedPID.Output/CurrentTargetSum) \* CurrentFullPower \*(1 + GameInfo.remainPower \* 0.0625f);

CM4SpeedPID.Output = (CM4SpeedPID.Output/CurrentTargetSum) \* CurrentFullPower \*(1 + GameInfo.remainPower \* 0.0625f);

### 超级电容充电

然后是超级电容充电部分。之前提到的超级电容充电方案其实存在一个缺陷，就当机器人把超级电容的电压低于一定值时，恒流恒压模块的电流基本流向超级电容，底盘电机的没有多少电流可以使用，也就会出现机器人爬完坡后会疲软。如果要改善这种情况的话，比较直接的方法就是拼命加大超级电容容值（*P.S. 土豪式解决方案*）。当然还有别的办法。

可以设计一个数控电源，对输出电流进行闭环控制，根据电机负载所需的电流对电容充电电流进行调配。在电机电流较小时可以增大给电容充电的电流，在电机满负荷时可以减小给电容充电的电流，这样就可以避免电容没电时，供给给电机的能量不足的情况。不过具体怎么设计电源以及如何将电源、电容、电机接起来这个我就没想出来。

还有就是增大超级电容的可利用的能量，我们实际上使用超级电容的能量是从电容电压从26.2V降到22.2V的能量，可以考虑想办法提高给超级电容充电的电压，就可以增加许多可利用的能量。我能想到的是给电容充电用升压模块，放电用降压模块，不过这个方法会存在效率低的问题，希望有dalao说一个更好的解决方案。

## 后记

我们队伍当前使用的方案如上述所说，是存在许多缺陷的，如果有什么建议和意见，希望可以多多指教。同时希望有队伍可以开贴说一下自己的功率限制方案或者相关优化方案，谢谢。