

底盘功率限制

version1.2.0authordungloi

理论

RoboMaster 对抗赛相关规则针对步兵、英雄和哨兵三个兵种设置了底盘功率限制，机器人底盘需在功率限制范围内运行；裁判系统持续监控机器人底盘功率，对功率超限的机器人按一定规则进行惩罚。考虑到机器人在运动过程中难以准确控制瞬时输出功率，为避免因瞬时超功率导致的惩罚，规则设置了缓冲能量 Z 。

更多描述请参考最新规则。

因此，底盘功率限制算法的目标是：在保证底盘运动可控的前提下，将底盘总功率（所有轮组叠加之和）限制在功率上限附近，并充分利用缓冲能量，实现底盘的平稳迅速启停、流畅运动，确保不受到底盘功率超限惩罚。

数据说明

对于机器人底盘，功率相关的数据来源有裁判系统、超级电容功率控制板量测和电机电调反馈，能够读取以下信息：

数据名	符号（单位）	数据来源	频率（Hz）	描述
底盘功率上限	$P_m(\rm W)$	裁判系统	10	当前等级下该兵种的底盘功率上限（max），超限则首先扣除缓冲能量
底盘缓冲能量	$Z(\rm J)$	裁判系统	10	缓冲能量余量，随裁判系统功率量测动态变化，若耗尽将导致超功率惩罚。目前（2023赛季）无增益上限为 60 J，触发飞坡增益后增加至 250 J 一次
底盘功率	$P_r(\rm W)$	裁判系统	10	裁判系统（referee）测得的底盘总功率
底盘功率	$P_c(\rm W)$	超电主控	1000	超电主控（cap）测得的底盘总功率
电机转速	$\Omega(\rm rpm)$	电调	1000	电机转速反馈

数据名	符号 (单位)	数据来源	频率 (Hz)	描述
电机 转矩 电流	$i_q(\rm A)$	电调	1000	电机转矩电流反馈，手册未指明量纲，经测试与输入一致：C620 电调原始值范围 $[-16384,16384]$ ，线性映射到 $[-20,20] \rm A$ 。

人为指定或计算得到以下数据，频率为 1KHz：

数据名	符号 (单位)	描述
底盘功率 模型计算 值	$P_{\rm model}(\rm W)$	根据功率模型计算得到的底盘功率
底盘功率 估计	$\hat P(\rm W)$	滤波后验估计结果
底盘功率 模型预测 值	$P_{\rm pred}(\rm W)$	根据电机功率模型计算得到，预测（predict）下一控制周期的底盘功率
动态底盘 功率上限	$P_{\rm ref}(\rm W)$	计算得到的动态变化、用于功率限制的底盘功率目标上限
底盘缓冲 能量目标 值	$Z_{\rm ref}(\rm J)$	缓冲能量目标值，用户给定
电机转速 目标值	$\Omega_{\rm ref}(\rm rpm)$	电机转速目标值，用户给定
电机转矩 电流目标 值	$i_{\rm ref}(\rm A)$	电机转矩电流目标值，基于转速目标值由控制器计算得出。C620 电调原始值范围 $[-16384,16384]$ ，线性映射到 $[-20,20] \rm A$ 。

电机功率模型

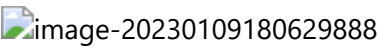
根据现有能够获取的数据，一个较为准确的电机功率模型为：

$$P = \Omega \cdot M + P_{\rm loss}$$

其中 M 表示输出转矩， $P_{\rm loss}$ 表示除机械输出以外的功率损耗，主要为铜损耗。


考虑电磁转矩和转矩电流 i_q 近似成正比（如下图，C620 电调搭配 M3508 电机速度闭环控制的电机性能曲线），定义转矩系数 $k_{\rm M}$ ，有

$$M = k_{\rm M} \cdot i_q$$



根据手册，转矩常数为 $0.3 \rm N \cdot m/A$ ，可作为 $k_{\rm M}$ 的初值参考。

使用转矩电流近似表示线电流，基于 $P_{\text{loss}} = R \cdot i^2$ ，在电机堵转的情况下统计 i_q 和功率观测 P_{measure} ，拟合损耗曲线。以 C620 电调 + M3508 电机为例，堵转电机，在 5000ms 内以 $[0:16384/2000:16384]$ 扫描转矩电流， $P-i$ 曲线拟合结果如下（供参考）：

 3508_total_fit_ref

其中 $\hat{R} = p_1$ ； p_2 视为 0； p_3 为电机所接入电路的静息功率。

功率估计

融合估计

对实际功率进行后验估计时，采用基于模型的先验预测数据（见下文）与量测数据通过卡尔曼滤波融合的形式取得：

预测： $\hat{x}_k^- = \mathbb{P}_{\text{model}} \hat{P}_{k-1}^- = \mathbb{P}_{k-1}^- + Q$ 更新： $K_k = \mathbb{P}_{k-1}^- (P_{k-1}^- + R)^{-1}$ $\hat{x}_k = \hat{x}_{k-1}^- + K_k (\mathbb{P}_{\text{measure}} - \hat{x}_k^-)$ $\hat{P}_k = (I - K_k) \mathbb{P}_{k-1}^-$

- **若底盘上未装有超级电容模块：** \mathbb{P}_r 量测（10Hz）与 $\mathbb{P}_{\text{model}}$ （1kHz）融合进行功率估计。
- ****若底盘上装有超级电容模块：**裁判系统不再直接监测底盘输出功率，此时 \mathbb{P}_c 量测（1kHz）与 $\mathbb{P}_{\text{model}}$ 融合进行功率估计。

 fusion.drawio

功率计算及预测模型

考虑 C620 电调具有优良的转矩电流调控性能，认为实际转矩电流能够密切跟随给定的转矩电流目标值。而电机的转速是缓慢变化的过程，不能突变，1ms 控制周期内的变化忽略不计。给出电机目标转速后，使用下式预测一个控制周期（1ms）后单个电机的功率：

$$P_{\text{pred}_j} = \hat{k}_{\text{M}} \cdot \Omega_j \cdot i_{\text{ref}_j} + \hat{R} \cdot i_{\text{ref}_j}^2$$

本算法**要求使用纯比例控制**计算电机的目标转矩电流 i_{ref} ，按照各电机的 PID 控制器参数，比例系数分别为 K_{p_j} ，同时进行结果的量纲转换（转换至 A）和限幅：

$$i_{\text{ref}_j} = \text{TO_A} [\text{LIMIT_MAX} \cdot (K_{\text{p}_j} (\Omega_{\text{ref}_j} - \Omega_j))] \text{ } \$$$

对于底盘的全部电机以及底盘的静息功率 P_0 ，叠加以预测总功率：

$$P_{\text{pred}} = \hat{k}_{\text{M}} \sum_j \Omega_j \cdot i_{\text{ref}_j} + \sum_j \hat{R} \cdot i_{\text{ref}_j}^2 + P_0$$

其中 \hat{R} 参数为拟合得到的常量；由于 k_{M} 并不能精确得到，需要根据上面的功率后验估计，基于模型不断进行修正：


$$\hat{k}_{\text{M}} = \frac{\hat{P} - \sum_j \hat{R} \cdot i_{\text{q}_j}^2 - P_0}{\sum_j \Omega_j \cdot i_{\text{q}_j}}$$

需要注意，裁判系统无法量测负功率，因此当以裁判系统作为功率量测且估计功率为负时，不应更新 \hat{k}_{M} 。

进行功率估计是为了得到更准确的预测模型，我们使用该模型进行下文所述的功率限制。

功率限制

算法主要流程如下：

 power_limiter.drawio

此外，注意以下特殊情况：

- 单个电机期望输出电流超限，需要按照电流上限解出其转速限制系数
- 当出现异常情况导致缓冲能量过低（超电电压过低）时，触发保护机制，应给出极低的目标功率
- 若出现裁判系统或超电主控断联，需做对应处理。

计算动态功率上限

算法中额外计算动态底盘功率上限 P_{ref} （而不直接使用 P_m 作为上限），一是为了应对电机启动阶段瞬时功率较大的情况，此时功率限制应适当放宽，若以 P_m 为实际功率上限将难以启动；二是为了充分利用部分缓冲能量。计算过程如下：

- 求取缓冲能量调节控制量 $u(Z)$ ，采用 PD 控制以增加稳定性：

$$\begin{aligned} e(Z) &= Z_{ref} - Z \\ u(Z) &= K_{pz}e(Z) + K_{dz}\frac{\Delta e(Z)}{\Delta T} \end{aligned}$$

- 参照裁判系统的缓冲能量计算逻辑 $Z = Z + (P_m - P)$ ，要使控制量 $u(Z)$ 生效，功率目标值应为：

$$P_{ref} = P_m - u(Z)$$

- 安全起见 Z_{ref} 一般设置为 20rm J 左右。为充分利用缓冲能量，对于不同的 P_m 应采用动态参数，比例系数的选取可参考下式：

$$K_{pz} = \frac{P_m}{Z_{ref}}$$

求解转速削减系数

由功率预测模型，当预测出的底盘功率大于动态功率上限时，同比例削减各电机目标转速。设置削减系数 k_l ，削减后电机 $\Omega_{ref}' = k_l \cdot \Omega_{ref}$ 。求取 k_l ，使得

$$\hat{k}_M \sum_j \Omega_j \cdot K_{pj} (k_l \Omega_{ref,j} - \Omega_j) + \sum_j \hat{R} \cdot [K_{pj} (k_l \Omega_{ref,j} - \Omega_j)]^2 + P_0 \triangleq P_{ref} \tag{1}$$

记

$$\begin{aligned} \alpha &= \hat{R} \sum_j K_{pj}^2 \Omega_{ref,j}^2 \\ \beta &= \sum_j (\hat{k}_M K_{pj} - 2\hat{R} K_{pj}^2) \Omega_j \Omega_{ref,j} \\ \gamma &= \sum_j (\hat{R} K_{pj}^2 - \hat{k}_M K_{pj}) \Omega_j^2 + P_0 - P_{ref} \end{aligned}$$

若解存在 ($\beta^2 - 4\alpha\gamma \geq 0$)，解得 $k_l = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$ ，取解 $\in (0, 1)$

若出现解不存在的情况，求使得 (1) 式左边结果最小的 k_l 值，以尽可能地限制功率： $k_l = -\frac{\beta}{2\alpha}$ 若 $K_{pj} (k_l \Omega_{ref,j} - \Omega_j)$ 超出电机可输出的转矩电流上限，则应按照该上限额外计算一个转速限制 $k_{e,j}$ ，使得 $K_{pj} (k_{e,j} k_l \Omega_{ref,j} - \Omega_j) \triangleq$ 对应超出的上限 并按照所有超限轮组中的 $\min\{k_{e,j}\}$ ，作用于底盘的全部轮组。经削减的轮组目标转速表示为：

$\Omega_{ref} = \min\{k_{ej}\} \cdot k \cdot \Omega_{ref}$ 经证明，削减转速的整个控制过程均能达到降低功率的效果。

快速开始

组件源码仓库地址: <https://github.com/ZJU-HelloWorld/HW-Components>

底盘功率限制算法依赖于卡尔曼滤波器和 PID 控制器。要在项目中使用该组件，需添加仓库内的以下文件：

```
algorithms/power_limiter.c
algorithms/power_limiter.h
algorithms/pid.c
algorithms/pid.h
algorithms/filter.c
algorithms/filter.h
tools.h
system.h
```

使用前准备

底盘功率限制算法涉及 CMSIS-DSP 矩阵运算等操作，使用前需要做以下准备：

- 添加源文件, 包含头文件路径；注意 DSP 版本须在 1.10.0 及以上
- 添加预处理宏以开启浮点运算单元 (FPU)
- 在使用 STM32CubeMX 生成项目时，请在 Code Generator 界面 Enable Full Assert，来帮助断言算法中的错误；在 main.c 中修改 assert_failed 函数以指示断言结果
- 在 system.h 中 system options: user config 处进行系统设置

示例

首先在限制器头文件 power_limiter.h 中设置是否安装超级电容模块的宏定义开关，例如：

```
/* USER CONFIG -----*/
#define SUPER_CAP_EXISTING 1
```

在项目中引用头文件：

```
#include "power_limiter.h"
```

实例化一个底盘功率限制器，并初始化一个静态参数结构体数组，参数含义见组件说明。如：

```
PwrLimiter_t limiter;

PwrLimitStaticParams_t static_param = {
    .motor_nums = 4u,
```

```

        .z_ref = 20u,
        .z_danger = 10u,
        .r_loss = <USER_CONFIG>,
        .p_bias = <USER_CONFIG>,
        .PwrKfParams = {
            .x = <USER_CONFIG>,
            .P = <USER_CONFIG>,
            .Q = <USER_CONFIG>,
            .R = <USER_CONFIG>,
        }
    };

```

其中 `r_loss`, `p_bias` (即 `P_0`) , `FusionKfParams` 参数与算法原理相关, 需要事先拟合、计算及整定。

初始化底盘功率限制器, 如:

```
PwrLimiterInit(&limiter, &static_param);
```

调用限制器时, 首先设置动态参数数组, 然后调用 `updateRuntimeParams` 方法进行数据更新, 如:

```

float spd_measure_rpm[<WHEEL_NUM>] = {...};
float iq_measure_[<WHEEL_NUM>] = {...};
float spd_ref_rpm[<WHEEL_NUM>] = {...};
Pid_t* motor_pid_instance[<WHEEL_NUM>] = {...};

PwrLimitRuntimeParams_t runtime_par = {
    .is_referee_online = true,
    .p_rfr_max = <REFEREE_DATA>,
    .z_rfr_measure = <REFEREE_DATA>,
    .p_rfr_measure = <REFEREE_DATA>,
    #if SUPER_CAP_EXISTING
    .is_super_cap_online = true,
    .super_cap_mode = PWR_LIMIT_SUPER_CAP_OFF,
    .p_dummy_max = <REFEREE_DATA>,
    .z_dummy_measure = <REFEREE_DATA>,
    .p_cap_measure = <REFEREE_DATA>,
    #endif
    .iq_measure_a = iq_measure_a_,
    .spd_measure_rpm = spd_measure_rpm_,
    .spd_ref_rpm = spd_ref_rpm_,
    .motor_pid_instance = motor_pid_instance_
};

limiter.updateRuntimeParams(&limiter, &runtime_par);

```

其中若底盘上安装了超电模块, 则启用 `is_super_cap_online` 等参数 (已由 `SUPER_CAP_EXISTING` 宏开关确定) 。

然后计算限制后的各电机转速，存储于用户提供的数组中。

注意：最终得到的是与动态参数设置顺序对应的所有电机的目标转速，还需要自行使用控制算法得到目标转矩电流，再发送给电调。

```
float limited_spd_ref_rpm[<WHEEL_NUM>];
limiter.calcLimitedSpd(&limiter, limited_spd_ref_rpm);
```

组件说明

PwrLimiter 类

底盘功率限制器。

属性

名称	类型	描述
static_params	PwrLimitStaticParams_t	功率限制静态参数，包括电机数量和用户设定的参数
runtime_params	PwrLimitRuntimeParams_t	功率限制动态参数，包括连接状态、裁判系统数据、转速及电流量测和目标值、电机 Kp 等
data	PwrLimitData_t	运行时中间项数据
p_ref_pid	Pid_t	控制缓冲能量的 PD 控制器
pwr_kf	Kf_t	融合功率模型计算值和观测值的卡尔曼滤波器

方法

名称	参数说明	描述
updateRuntimeParams	传入动态参数数组指针 PwrLimitRuntimeParams_t* params	更新运行时动态参数
calcLimitedSpd	传入数组首地址 float* limited_spd_ref_rpm，用于存储限制后电机转速的输出结果	计算限制后的各电机转速

PwrLimitStaticParams 结构体

存储静态参数。

名称	类型	示例值	描述
motor_nums	uint8_t	4	电机数量
z_ref	uint16_t	20	缓冲能量目标收敛值
z_danger	uint16_t	10	缓冲能量最小危险值，以避免缓冲能量耗尽

名称	类型	示例值	描述
p_bias	float	8.0f	底盘静息功率
r_loss	float	/	功率损耗相关参数
PwrKfParams	/	/	融合功率模型计算值和观测值的卡尔曼滤波器参数

PwrLimitRuntimeParams 结构体

存储算法运行时所需的动态参数。

名称	类型	示例值	描述
is_referee_online	bool	true / false	裁判系统连接状态
is_super_cap_online	bool	true / false	(仅当安装超电模块时启用) 超级电容主控连接状态
super_cap_mode	PwrLimitCapMode_t	PWR_LIMIT_SUPER_CAP_OFF PWR_LIMIT_SUPER_CAP_NORMAL PWR_LIMIT_SUPER_CAP_BOOST	(仅当安装超电模块时启用) 超级电容状态模式
p_rfr_max	uint16_t	50	裁判系统功率上限
z_rfr_measure	uint16_t	60	裁判系统缓冲能量量测
p_rfr_measure	float	50.0f	裁判系统功率量测
p_dummy_max	uint16_t	50	(仅当安装超电模块时启用) 超级电容启用时, 提供的伪功率上限
z_dummy_measure	uint16_t	60	(仅当安装超电模块时启用) 超级电容启用时, 提供的伪缓冲能量
p_cap_measure	float	50.0f	(仅当安装超电模块时启用) 超级电容主控功率量测
iq_measure_a	int16_t*	/	所有电机转矩电流量测的数组首地址, 单位: $\rm A$
spd_measure_rpm	int16_t*	/	所有电机转速量测的数组首地址, 单位: $\rm rpm$

名称	类型	示例值	描述
spd_ref_rpm	int16_t*	/	所有电机转速目标值的数组首地址，单位：\$\\rm rpm\$
motor_pid_instance	Pid_t**	/	底盘电机转速控制器句柄列表指针

附录

版本说明

版本号	发布日期	说明	贡献者
<div>version1.0.0</div>	2021.12.06	完成新版功率限制研发，采用简单模型	薛东来
<div>version1.0.1</div>	2022.07.17	配合新超电逻辑联调，整理代码	薛东来
<div>version1.1.0</div>	2023.01.11	修改估计细节，完善模型	薛东来
<div>version1.2.0</div>	2023.05.17	1. 拟合曲线 2. 测试并修正电机输出电流超限问题以及km估计不准问题 3. 结合超电逻辑完善	薛东来 赵炜