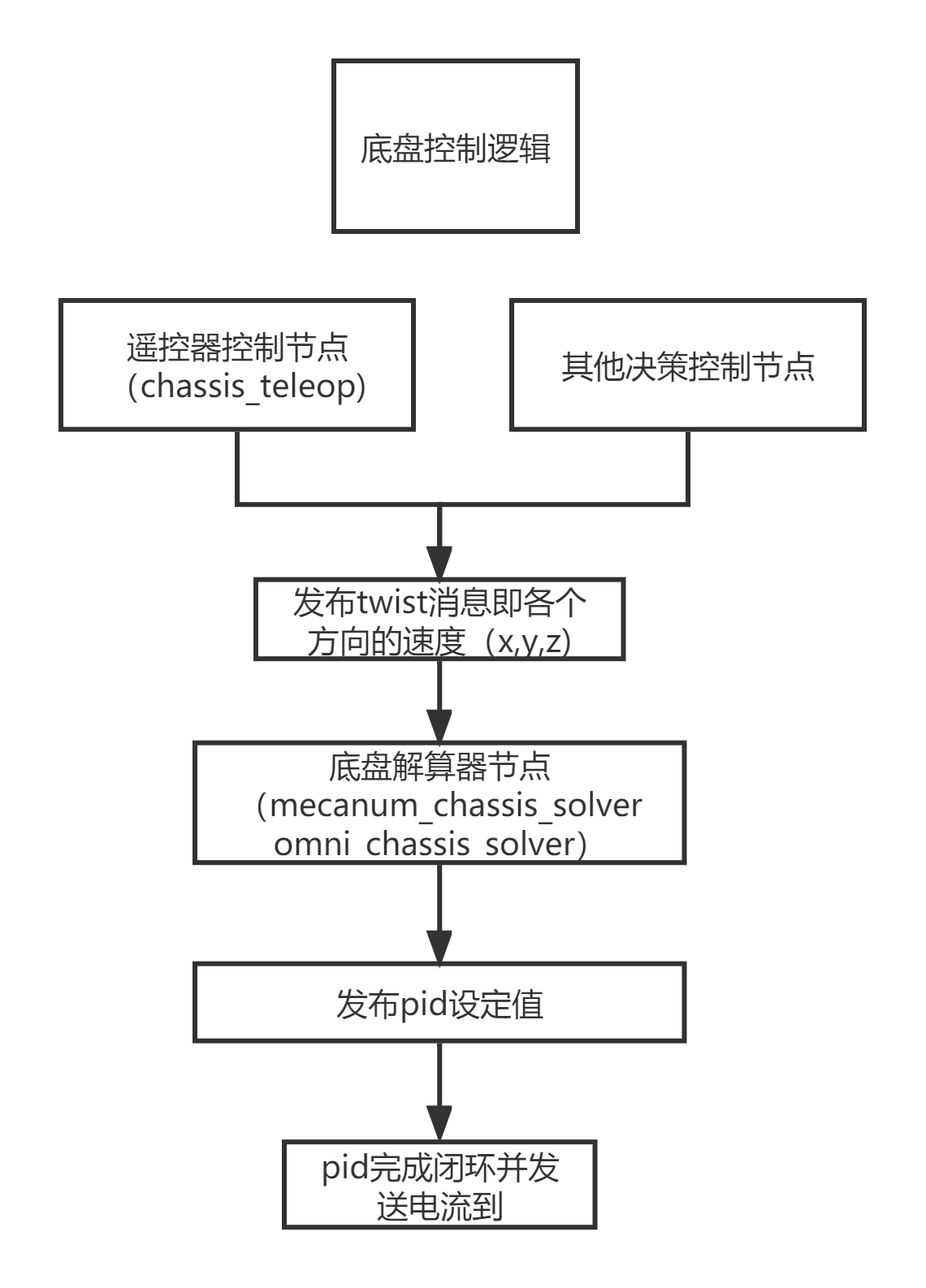
****

**底盘逆运动学解算**

全向轮底盘是一种可以在平面内任意方向上移动的底盘，通常采用三个或四个全向轮作为底盘的驱动装置。与传统的差动驱动底盘不同，全向轮底盘的驱动方式需要解决底盘的逆运动学问题，即给定底盘的速度和方向，如何控制每个轮子的转速来实现底盘的运动。

哨兵机器人的全向移动算法

解算公式与麦轮相似，但因为全向轮安装方向与麦轮不同且全向轮底盘为圆。因此各项速度的系数有所改变。本次哨兵采用四轮全向轮

在全向轮底盘中，通常使用笛卡尔坐标系来描述底盘的位置和速度，其中x和y表示底盘在平面内的位置，θ表示底盘的朝向角。底盘的速度可以表示为Vx、Vy和Vz三个分量，分别表示底盘在x、y和θ方向上的速度。对于一个四个全向轮的底盘，它们的轮子可以分别表示为RF、LF、LB和RB，每个轮子的速度可以表示为Vrf、Vlf、Vlb和Vrb。

根据全向轮的运动学特性，我们可以推导出轮子速度和底盘速度之间的关系式，这些关系式被称为正解算公式和逆解算公式。正解算公式是指根据底盘速度和方向计算出每个轮子的速度，而逆解算公式是指根据每个轮子的速度计算出底盘的速度和方向。以四个全向轮底盘为例，全向轮的轮子安装在底盘的四个角落处，每个轮子都可以在x、y、z方向上提供力矩，控制底盘的运动。在逆解算公式中，我们需要通过轮子速度来计算底盘速度和方向，计算方法如下：

全向轮逆运动解算：

= (-+

= (++

= (+++

= (-++

全向轮正解算公式：正运动学解算则相反，它是指在已知各个执行机构的控制量时，计算机器人在空间中的实际位置和方向。在4轮全向轮底盘中，正运动学解算可以计算出机器人的位置和方向，从而实现对机器人的控制和导航。(以向前为x正方向 向左为y负方向 为）

= (-+

= (++

= (+++

= (-++

**底盘里程计及IMU联合优化**

首先就我们为什么需要机器人实现里程计功能的分析

定位需求：机器人需要实时获得自身的位置信息，以便在环境中准确定位并执行任务。对于赛场上哨兵的准确定位来说，此时，里程计成为一种非常重要的定位方式。里程计可以根据机器人轮胎的转动情况，计算机器人的位姿，从而实现高精度定位。

自动路线规划需求：自动路线规划是指机器人自主选择路径以达到目标地点的功能。机器人需要实时了解自己的位置和目标位置，从而规划合适的路径。里程计正好可以为机器人提供位置信息，帮助机器人进行自主导航和路径规划。通过结合其他传感器数据，如激光雷达、相机等，机器人可以进行更加精确的定位和路径规划。对于哨兵机器人的功能实现做出了基础。

首先通过电机数据以及逆解算公式获全向轮正解算公式来提供实际哨兵的速度。

= (-/4/

= (-/4/ \*

= (-/2/ \*

通过旋转速度计算出旋转的角度，并使底盘的和速度方向映射到底盘旋转后的速度方向。再使用ros2中rclcpp库中的get\_clock()来获取当前是时间，并与上一时间计算出时间差Δt。（通过rclcpp::Clock::now()函数，它返回一个rclcpp::Time类型的时间戳对象。可以调用该函数多次来获取多个时间戳，然后通过计算两个时间戳之间的差异来计算出两次获取时间戳的时间差Δt）根据我们获取的就可以计算出初始启动坐标下的位移X = \*Δt, Y = \* Δt。

里程计的计算是指以机器人上电时刻为世界坐标系的起点（机器人的航向角是世界坐标系X正方向）开始累积计算任意时刻机器人在世界坐标系下的位姿。通常计算里程计方法是速度积分推算：==通过左右电机的编码器测得机器人的左右轮的速度VL和VR，在一个短的时刻△t内，认为机器人是匀速运动，并且根据上一时刻机器人的航向角计算得出机器人在该时刻内世界坐标系上X和Y轴的增量，然后将增量进行累加处理，然后根据以上描述即可得到机器人的里程计。

联合IMU优化部分：

在全向轮正向解算以至于哨兵的2D位姿较为精准的情况下，可以清楚哨兵位于初始位置的哪个位置。然而，由于编码器误差、轮子打滑、长时间误差累积等，仅通过哨兵三种速度的计算最终一定会出现先较大误差。因此IMU可以提供机器人运动中的姿态信息，通过姿态信息结合编码器信息，可以更精确地计算机器人在三维空间中的位移、旋转和速度等信息，从而大大提高了机器人里程计的精度。，使用IMU获取的yaw值作为航向角来进行优化。或者激光雷达可以获得以下各种数据：

1. 通过激光雷达观测数据与地图进行匹配，估计出机器人的位姿；
2. 第二，当机器人估计出较准确的位姿时，通过激光雷达的观测数据建立环境地图；

第三，在机器人导航过程中，检测地图中的未知障碍物。这些数据可以被用来进一步优化里程计模型，提高哨兵位姿的准确性。

**底盘控制优化**

当机器人底盘通过遥控器控制进行前进时，由于遥控器手柄存在的不稳定性和人的手部抖动等因素，导致发送给机器人底盘的指令信号可能存在噪声和突变。这些噪声和突变会导致底盘运动不平稳，影响机器人的行进方向和速度控制。

机器人底盘遥控器信号过滤，使用一阶卡尔曼滤波器有以下优势：

1实时性好：一阶卡尔曼滤波器的计算量相对较小，实时性好。可以在实时控制中使用，对于需要快速响应的机器人运动控制任务尤为适用。

2适应性强：一阶卡尔曼滤波器可以自适应地调整滤波器参数，根据实时的传感器测量值和模型预测值，自动更新滤波器状态，使得滤波器更加准确地估计机器人的运动状态。从而更好的适配不同型号机器人

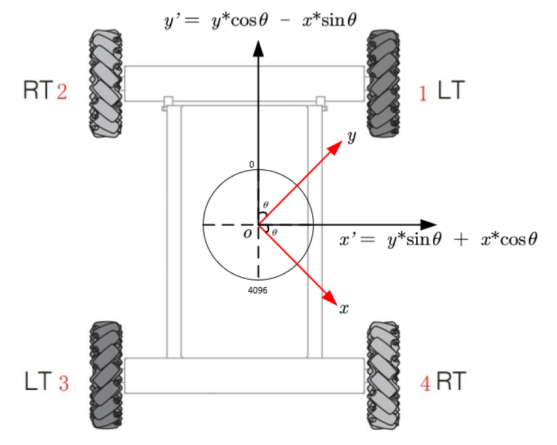
3抗噪声能力强：机器人运动过程中可能会受到各种噪声的干扰，如传感器误差、机器人运动震动等。一阶卡尔曼滤波器能够通过对噪声进行滤波，减小噪声对机器人运动状态估计的影响。从而应对复杂环境

**底盘跟随云台及小陀螺**

底盘跟随云台及小陀螺可以增加机器人的稳定性和灵活性，提高其在复杂环境中的操作能力。

当机器人需要在不平稳的地面上行驶时，底盘跟随云台可以帮助机器人保持平稳。底盘跟随云台能够感知云台的姿态和运动状态，并对底盘进行相应的调整，使机器人能够保持平衡。从而更好应对赛场上的道路不平整环境。更好的做出机器方面调整。

小陀螺则可以帮助机器人提升防击打能力并且在转弯或者突然停止的时候保持平衡。小陀螺的惯性可以使其继续保持原有的运动状态，从而减少机器人的晃动和摆动。

1. 获取底盘与云台的相对角度θ：首先需要通过传感器等手段获取底盘的绝对角度和云台的绝对角度，然后计算它们之间的相对角度θ。底盘绝对角度可以通过测量YAW轴电机的角度来获取，云台绝对角度可以由YAW轴电机实时提供。
2. 将整车运动速度（大小和方向）分解到底盘坐标：根据相对角度θ，可以将整车运动速度（大小和方向）分解到底盘坐标系中。具体来说，可以通过使用三角函数计算出底盘在云台坐标系中的运动速度和方向。
3. 根据底盘坐标速度进行全向轮速度分解：接下来，可以根据底盘在云台坐标系中的速度和方向，使用全向轮的运动学模型计算出每个麦轮的转速，从而实现底盘的运动控制。
4. 整车效果表现为按云台坐标系运动：通过以上步骤，可以实现底盘跟随云台的效果。整车的运动效果将表现为按照云台坐标系中的方向和速度进行运动，从而实现底盘跟随云台的控制。

在底盘跟随云台模式下，需要使用PID算法来计算底盘应该旋转的角度。具体地，首先需要计算机器人上电时yaw电机编码器的值与底盘跟随云台角度时yaw电机编码器的差值，作为PID算法的设定值。接下来，需要实时读取yaw电机的position值（一种不会跳变的与电机编码值单位相同的数据）作为PID算法的实际值。通过对设定值和实际值进行比较，PID算法可以计算出底盘应该旋转的角度，然后将输出值赋给底盘的旋转速度，从而实现底盘跟随云台的效果。

在小陀螺模式下，直接设置不超过功率的底盘旋转速度即可，这是因为在该模式下，底盘旋转的目的不是为了跟随云台方向，而是为了保持平衡与避免遭到射击。因此，只需要限制底盘旋转的功率大小，就可以达到机器人的此项功能。

调参过程：

首先需要确定机器人底盘的重心位置，通过在装满弹丸的情况下与未装载弹丸的情况下进行机器人不同的小陀螺运动与实验。从而确定机器人的重心在其中发生的偏移

然后需要确定机器人在小陀螺模式下需要的最大转速和加速度。根据实际数据调整机器人的转速从而找到合适与便于移动与转向的机器人旋转速度

然后进行调整PID参数，以使机器人在小陀螺模式下保持稳定。PID控制器可以控制机器人的转速和转向角度，以保持机器人在行驶过程中的稳定性。

**底盘功率控制**

首先，需要将电机转速控制转换为电流控制。这可以通过使用电机转矩常数（或称为电机Kt常数）来实现，它定义了电机输出的转矩和输入电流之间的关系。在电流控制中，我们将输出电流设置为控制变量，并使用PID控制器来控制电流，从而控制电机的转速。

具体而言，电流控制方案包括以下步骤：

读取电机电流值：使用电流传感器读取电机的实时电流值。

计算误差：将期望电流值与实际电流值之间的差值作为误差，这是控制器需要调整的量。

计算控制器输出：使用PID算法计算控制器输出，其中包括比例项、积分项和微分项。

限制控制器输出：将控制器输出限制在可接受的范围内，以确保不会超过电机的最大电流限制。

输出控制信号：将控制器输出转换为电机驱动器的控制信号，从而控制电机的电流。底盘电机功率控制算法

超级电容部分

我们队本赛季首次采用超级电容模组，并使用了两种不同的拓扑结构。我们研发出一套串级底盘功率控制算法，适用于电容与底盘并联或非并联的拓扑结构，以实现底盘输入功率的合理分配和控制。该算法通过稳定电容容量，实现了超级电容输出的主动开启或关闭，并且无论电容电量如何，底盘裁判系统输出功率都优先响应操作手的指令。

ω 缓冲能量闭环 根据裁判系统反馈数据的功率上限和当前缓冲能量，计算出当前最大底盘输入功率， 发送给超级电容控制板，将缓冲能量稳定在 焦耳。该闭环可以防止长时间高功 率输入的控制误差对缓冲能量造成的连续消耗，从而避免超功率。同时可以迅速使 用飞坡增益的高额缓冲能量，以快速恢复电容电量。 ω 电容电量闭环 根据超级电容原理

𝑃𝑖𝑛 = 𝑃𝑐𝑎𝑝 + 𝑃𝑐ℎ𝑎𝑠𝑠𝑖𝑠

在输入功率不变的情况下，可以通过主动控制底盘电机功率输出，来改变电容的充 放功率。因此我们通过限制四个底盘电机的电调发送电流值，来限制底盘电机功率 输出，最终实现电容容量控制。 为了提高该环的响应速度，我们让底盘电机输出功率在底盘输入功率上限的基础上 进行浮动。根据测试结果，在计算初始浮动值时，我们简单的认为 通信发送 值 与底盘功率有如下关系。 𝑁𝑐𝑚𝑑 = 𝑃𝑚𝑜𝑡𝑜𝑟 × 250 该环反馈值为当前电容容量或电容电压，设置值为目标电容容量或电容电压，输出 值为 通信发送值的变化量，通过 控制将电容容量稳定在目标值。 最终，我们得到了缩放系数

𝑘 = 𝑃𝑖𝑛 × 250 − 𝑃𝐼𝐷𝑜𝑢𝑡 ∑𝑁𝑐𝑚𝑑

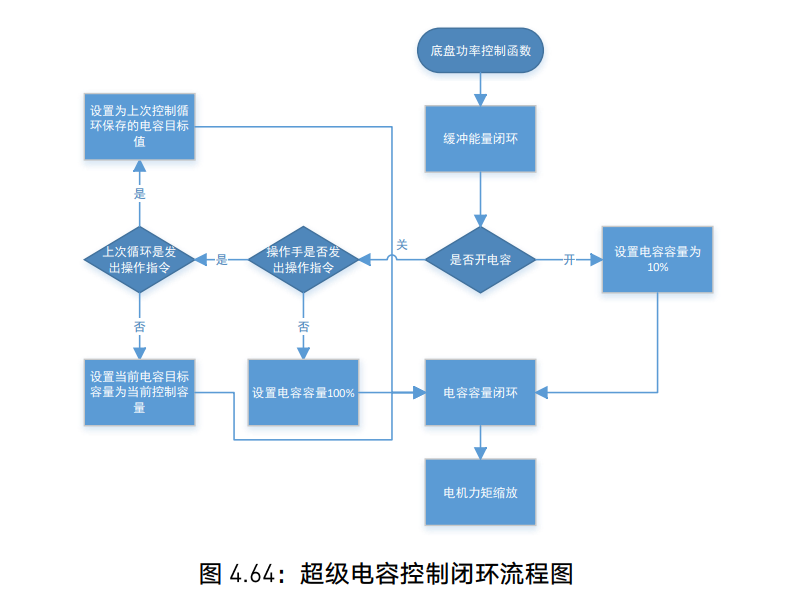
注：电机输出减小，电容容量变大，公式中注意 输出符号为负。

给电机的最终指令为

𝑁𝑐𝑚𝑑 ′ = 𝑘𝑁𝑐𝑚𝑑，

在 大于 时不进行缩放。 战队官方文档 该环实现效果为，可以将电容容量稳定在任意大小，当设定值从高容量，设定为低 容量时，即为开启超级电容，电容额外放电，容量变小并最终稳定；当设定值从低 容量设定为高容量时，电容充电，容量增大并最终稳定。 ω 动态电容目标值设定 分区赛使用电容与底盘并联的拓扑结构，将电容目标值设定为两个固定值，在开启 电容时，电容电压设定在 ，关闭时电容电压设定为电池电压 。但是简单的 两个目标值会出现，当操作手在低电容容量选择关闭超级电容时，该算法会大幅度 限制底盘功率以将电容容量恢复至较高状态。此时操作手几乎会失去对于底盘的控 制。 (图 ：超级电容控制闭环流程图)

分区赛后我们使用动态的电容目标值以解决该问题。在开启电容时仍使用固定的低 容量目标值。 在关闭电容时，当操作手未发送移动指令，即未按下 ，小陀螺按键或主动刹 车按键，将选择一个高电容容量为目标电压值。 当操作手发送移动指令，将选择当前电容容量为目标电容值并保持该目标大小，直 到操作手停止发送指令。由于积分项响应速度较慢，不能每次控制循环都选择一个为了解决底盘功率受电容充电影响的问题，



我们在分区赛后采用了动态电容目标值的方法。在开启电容时，我们仍然使用固定的低容量目标值。但在关闭电容时，如果操作手未发送移动指令，则会选择一个高电容容量作为目标电压值。当操作手发送移动指令时，我们将选择当前电容容量作为目标容量，并将目标大小保持不变，直到操作手停止发送指令。由于积分项响应速度较慢，我们不能在每次控制循环中选择一个新的目标值，而应在连续运动时保持相同的目标值。此时电容将停止充电和放电，底盘功率将优先用于响应操作手指令。当操作手停止发送控制指令时，目标值将恢复为高容量。

采用动态电容目标值的方法可以保证底盘功率不会因电容充电而受到限制。同时，由于没有指令时电容容量为高容量值，该算法还可以实现一个动态的被动刹车功率控制：低电容容量时刹车功率小，高电容容量时刹车功率高。但由于基础功率浮动值受电池电压、底盘回路电压等因素影响较大，我们目前主要依靠积分项来稳定功率限制。对于电容与底盘并联的拓扑结构，在不同的底盘电压下，电机的输出功率与指令值的关系会有所不同。未来我们可以通过更精确的电机功率模型减少对积分项的依赖，以提高响应速度。我们还可以将底盘目标速度进行动态调整，减少对被动限制的依赖，从而间接提高响应速度，同时保证底盘精确移动。