基于ROS系统的RM步兵机器人设计

李云兮
湖北工业大学　湖北武汉　430068

摘　要：本文以原有Robomaster步兵机器人为基础，进行基于ROS系统的步兵机器人设计，使基于ROS的Jetson nano和PC承担更多的任务。本文设计了基于ROS软件框架的步兵机器人控制系统；搭建并测试了机器人样机，在Jetson nano上扩展了传感器模块（云台IMU）。结果证明实现了机器人基本功能，达到了设计要求。

关键词：ROS；多传感器；嵌入式系统；姿态解算

Design of RM infantry robot based on ROS system

Yunxi Li
Hubei University of Technology, Wuhan, Hubei, 430068

Abstract: Based on the original Robomaster infantry robot, this paper designs an infantry robot based on the ROS system, so that the ROS-based Jetson nano and PC can undertake more tasks. In this paper, an infantry robot control system based on the ROS software framework is designed; the robot prototype is built and tested, and the sensor module (PTZ IMU) is extended on the Jetson nano. The results show that the basic functions of the robot are realized and the design requirements are met.

Keywords: ROS; Multisensor; Embedded systems; Attitude solution

引言：

随着ROS作为机器人编程实际标准的做法被广泛采用，开发者和开放的领域也随之越来越多。Marin Plaza Pablo[1]等研究人员基于ROS的自动驾驶汽车研究平台中的全局和局部路径规划研究。RoboMaster机甲大师高校系列赛是由大疆创新发起的机器人赛事。步兵机器人是RM机器人序列中的重要一员，随着赛事迭代发展和新种类赛事的出现，对于步兵机器人的性能要求不管是从深度还是广度都越来越高。在现有条件和新的要求下，用机载PC承担更多的工作是一个合适的选择。

一、机器人系统设计

1.1 任务分析

原有步兵机器人由STM32做传感器的数据采集处理（除视觉）和底盘、云台，miniPC进行视觉识别处理并通过USB发送数据辅助STM32进行判断。本文以原有Robomaster步兵机器人为基础，进行基于ROS系统的步兵机器人设计，使基于ROS的NVIDIA Jetson nano和PC承担更多的功能。在功能实现方面，机器人需要完成以下功能，框架如图2.1所示。

（1）实现PC端控制机器人全向移动；

（2）行进中可实现小陀螺运动；

（3）实现多传感器下的机器人状态识别。

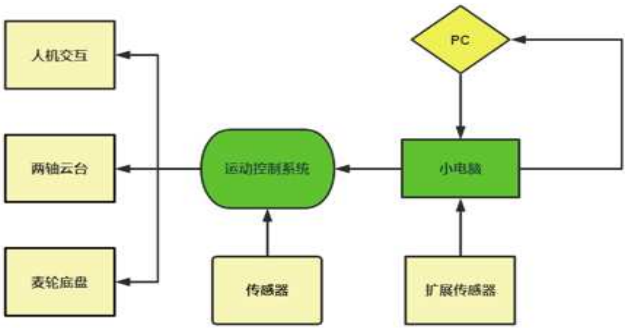

图2.1　机器人框架

1.2 机器人硬件平台设计与搭建

1.2.1 双轴云台和麦克纳姆轮底盘设计

（1）双轴云台的双轴指的是Pitch轴和Yaw轴，由两个执行机构分别控制Yaw轴航向角和Pitch轴俯仰角。样机搭建选择的是两个不同规格的HWZ020舵机分别控制Yaw轴航向角和Pitch轴俯仰角。其中，控制控制Yaw轴航向角的HWZ020舵机最大转角270°；控制Pitch轴俯仰角的HWZ020舵机最大转角180°。

（2）麦克纳姆轮（Mecanum Wheel），又称为艾隆

轮（Ilon Wheel）。麦克纳姆轮的布局主要分为X和O型，这里采用Robomaster的组装方式X型。样机搭建选择的麦克纳姆轮底盘由4组麦克纳姆轮组成。每组包括1个MG513P30-12V电机、1个霍尔编码器、1个60mm的麦克纳姆轮。

1.2.2控制器及传感器设计

除了底盘和云台，机器人还需要作上位机和下位机的控制器，以及用来测量各项数据辅助决策的传感器。

（1）上位机（主控制器）：主流工控机基本架构、使用方法大同小异，只是在性能上存在差别。NVIDIA Jetson Nano满足设计需求，本文选择其作为主控制器，主要负责IMU的采集、分析STM32发送的传感器数据以及和PC端的通讯。

（2）下位机（STM32）：运动控制系统的控制器。按照Robomater的套件标准，采用STM32F407VET6单片机核心板及扩展板组成，主要用于机器人运动控制系统的实现以及相关传感器数据的采集，同时有接口支持PS2、OLED、串口通信、IMU数据采集等功能模块接入。

（3）IMU：惯性测量单元（Inertial measurement unit）是测量物体三轴姿态角（或角速率）以及加速度的装置。为了进行机器人姿态解算，麦克纳姆轮底盘和双轴云台上各一个。麦克纳姆轮底盘上的IMU连接在STM32上，数据由STM32采集；双轴云台上的IMU直接与主控制器连接，开展基于ROS的主控制器传感器研究。

底盘IMU是集成在STM32底盘主板上的ICM20948。ICM20948是TDK推出的一种九轴IMU。云台IMU选择N100模块。N100模块是一款九轴姿态传感器，有Type-C接口，支持与ROS通信。

1.3机器人控制系统软件设计

本文基于ROS软件框架构建步兵机器人控制系统。

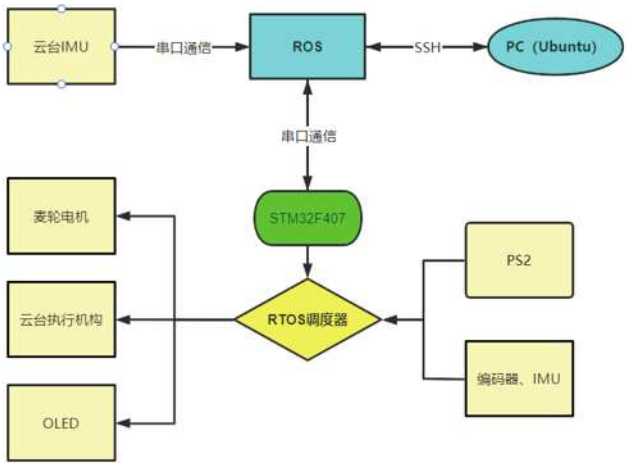

图2.2　软件设计框架

根据功能要求，构建的步兵机器人控制系统分为三层：PC端；基于工控机的机器人操作系统；基于STM32的运动控制系统，软件设计框架如图2.2。

1.4机器人控制策略设计

1.4.1姿态解算算法

机器人的姿态和航向显示了车体自身坐标与导航坐标系之间的相对位置关系。四元数法计算方便并且能更加合理地描述载体的刚体空间运动，被广泛应用。

龙格-库塔法（Runge-Kutta）是一种高精度单步四元数即时更新算法。一阶龙格-库塔法是用初始端点的斜率来近似T时间段内的斜率。套用到四元数中，传入参数为三轴陀螺仪的值，输出更新四元数，如式2.1。

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_{t+\Delta t} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} + \frac{\Delta t}{2} \begin{pmatrix} -\omega_x \cdot q_1 - \omega_y \cdot q_2 - \omega_z \cdot q_3 \\ +\omega_x \cdot q_0 - \omega_y \cdot q_3 + \omega_z \cdot q_2 \\ +\omega_x \cdot q_3 + \omega_y \cdot q_0 - \omega_z \cdot q_1 \\ -\omega_x \cdot q_2 + \omega_y \cdot q_1 + \omega_z \cdot q_0 \end{pmatrix} \quad (2.1)$$

1.4.2小陀螺算法

小陀螺行进，顾名思义就是在旋转的同时按照某个坐标系进行全向移动。程当机器人开始小陀螺运动时，按照底盘相对坐标系运动显然不能达到效果。在RoboMaster中，步兵机器人是由云台和底盘两个主要部分组成的。因此，按照实际需求，选择按照云台相对坐标系进行全向移动。

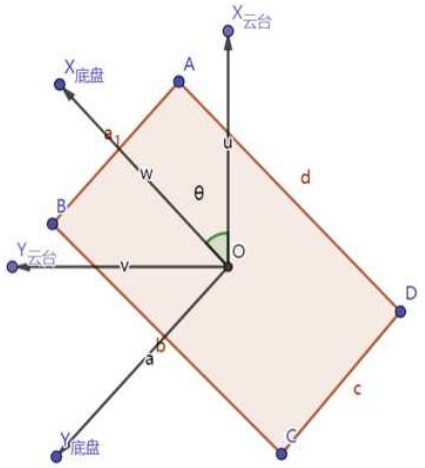

图2-3　底盘相对坐标系与云台相对坐标系示意图

如图2-3，计底盘相对坐标系与云台相对坐标系夹角θ，将云台相对坐标系的机器人运动速度（大小和方向）转换到底盘相对坐标系，如式2.2、2.3。

$$x_{底盘} = y_{云台}\sin\theta + x_{云台}\cos\theta \quad (2.2)$$

$$y_{底盘} = y_{云台}\cos\theta - x_{云台}\sin\theta \quad (2.3)$$

对底盘相对坐标系机器人速度进行麦轮速度分解，整车效果应表现为按云台相对坐标系运动。

1.4.3 PID算法

PID控制器是根据系统的误差，利用比例常数提高系统的调节精度、微分常数加快响应速度、积分常数消除稳态误差，从而计算出需要的控制量进行控制。由于计算机控制是采样控制。因此PID控制规律的实现必须用数值逼近的方法。通过模拟PID离散化变为差分方程，得到位置式PID（式2.4）增量式PID（式2.5），其中，e（k）是输入量，e（k-1）是上次输入量，out是输出量，sum：e（k）累计值。

$$out_{位置} = P * e(k) + I * sum + D * [e(k) - e(k-1)] \quad (2.4)$$

$$out_{增量} = P * [e(k) - e(k-1)] + I * e(k) + D * [e(k) - 2e(k-1) + e(k-2)] \quad (2.5)$$

二、机器人软件设计与实现

2.1 基于FreeRTOS的STM32程序框架

STM32的程序是基于FreeRTOS编写的。FreeRTOS对于设计目的来说最重要的优点是可读性和可移植性强。

本文根据需要完成的功能设计了7个任务："Balance_task"，"ICM20948_task"，"show_task"，"led_task"，"PSTWO_task"，"DATA_task"，"Control_task"。分别依次是机器人底盘运动控制任务、底盘IMU数据读取任务、OLED显示屏读取任务、LED闪烁任务、PS2手柄控制任务、传感器数据发送任务、云台运动控制任务。同时，ROS通过串口3发送的命令由相应中断响应程序处理，程序框架图如图3.1。

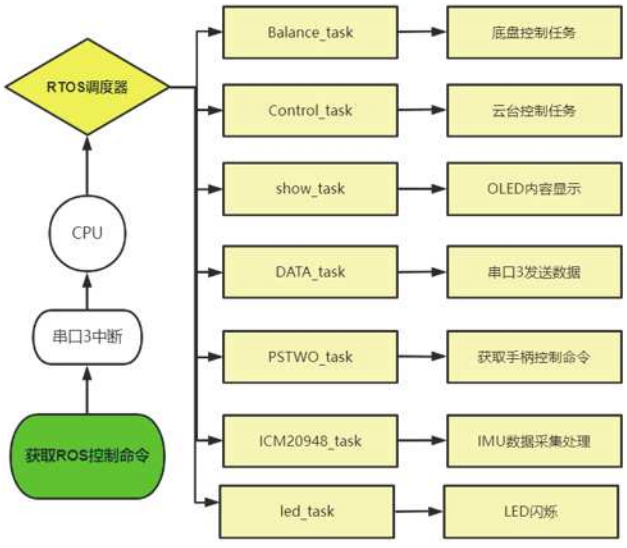

图3.1 基于FreeRTOS的STM32程序框架

2.2 机器人运动控制程序实现

2.2.1 底盘运动学实现

如上文程序框架，"Balance_task"任务是负责底盘的运动控制。在程序中，通过PS2或者PC键盘输入得到三轴目标速度后进行运动学逆解，得到四个麦轮目标速度，运动学实现逻辑如图3.2。

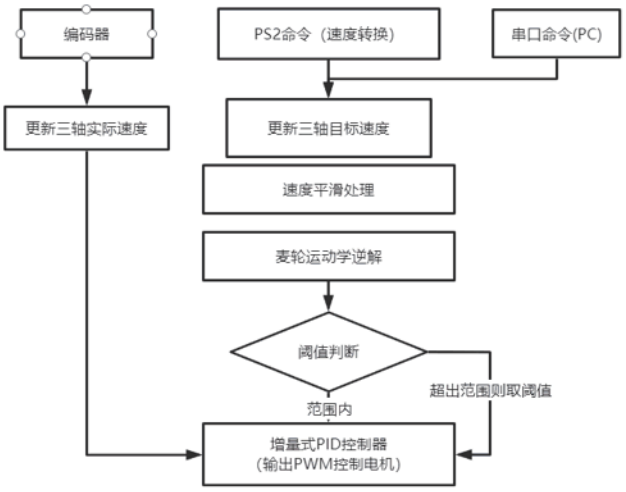

图3.2 运动学实现逻辑图

目标速度和实际速度有差值，因此使用2.4.3小节的PI增量式PID控制器，使得电机的实际输出速度趋近于目标值。

2.2.2 云台控制实现

PS2控制和上位机串口控制都是通过设置改变舵机目标角度来实现舵机控制的。目标角度最后要转换为目标PWM值控制舵机。云台控制实现流程如图3.3。为了更好地控制效果，这里需要用到位置式PID控制器，使得电机的实际输出速度趋近于目标值。

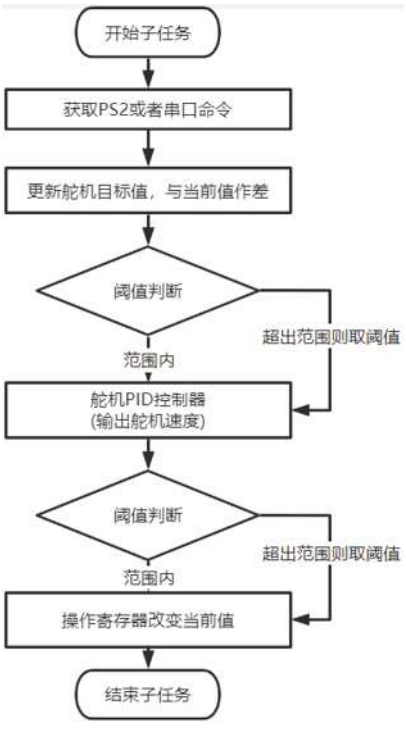

图3.3 云台控制实现流程图

2.3 姿态解算程序设计

2.3.1 底盘IMU姿态解算设计

ICM20948的原始数据需要进行零点漂移消除。航向角的解算使用了龙格-库塔法（Runge-Kutta）。初始化后，首先执行陀螺仪去除零点漂移，然后读取角速度原始数据。处理得到弧度制三轴角速度。代入公式更新四元数，更新后四元数取模、归一化，利用矩阵知识解算出姿态角。

2.3.2 云台IMU姿态数据读取

N100惯导模块默认使用的通信方式是串口通信。根据通信协议，可以通过ROS读取需要的IMU数据作为云台的姿态。由于N100惯导模块发送的数据类型有多种，且数据长度也不同，则需要分类接收数据。根据所接收数据帧的帧头和帧尾判断此刻接收到的数据是否有效，再由指令类别和数据长度判断数据的类型。

2.4 小陀螺行进实现

小陀螺行进的算法实现分两个部分，云台和底盘，实现效果应为底盘不断旋转，同时可以按照云台相对坐标系进行全向移动。

底盘部分：根据小陀螺算法原理，通过云台和底盘的两个IMU的数据，可以得到云台相对坐标系和底盘相对坐标系分别与参考坐标系的夹角 α 、β，进一步可得出两个夹角 α 、β 的差值，即底盘相对坐标系与云台相对坐标系夹角 θ 。根据2.4.2小节的推导结果，底盘表现为按云台相对坐标系运动。

云台部分：按云台相对坐标系运动时，在没有命令改变云台yaw轴航向角情况下，云台相对坐标系与参考坐标系的夹角 α 应保持不变。ROS从N100模块解算出姿态角并通过串口发送到STM32后，STM32根据N100的yaw轴航向角的前后差值，输出舵机PWM值控制，利用PID控制器控制夹角 α 的值的稳定。

在硬件选型中，机器人样机的云台底部控制舵机的最大转角270° ，不能实现在麦轮底盘自旋转时保证云台

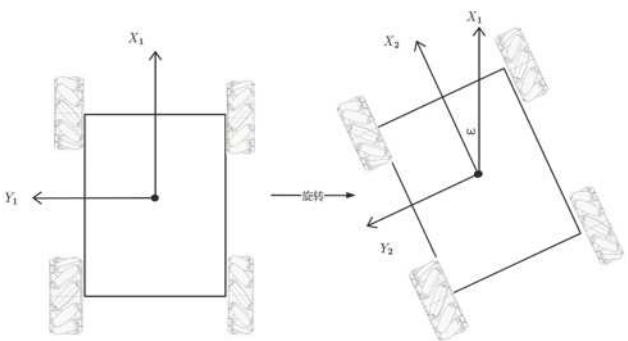

图3.4　四元数解算程序实现

yaw轴航向角不变。因此提出替代实现方案。以机器人初始上电位置的底盘相对坐标系为小陀螺行进的参考坐标系。此时的 θ 即为底盘IMU解算出的yaw轴航向角 ω 。以底盘IMU解算出的航向角 ω 为入口参数，如图3.4所示，执行小陀螺程序，效果应表现为机器人自旋转的同时可按初始位置的底盘相对坐标系全向运动。

2.5 机器人通信设计与下位机人机交互实现

2.5.1 下位机人机交互实现

底盘控制系统的人机交互由OLED显示和PS2遥控两部分组成，在RTOS调度器中是”show_task”和”PSTWO_task”两个任务。通过OLED显示和PS2遥控，实现逻辑如图3.5，实现在没有ROS命令的时候接收PS2控制命令。

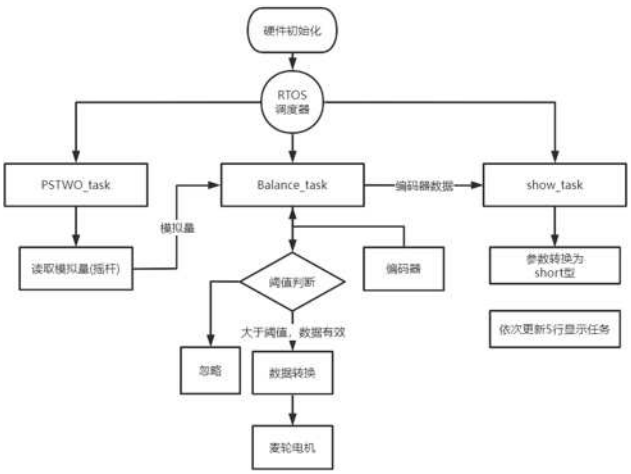

图3.5　人机交互实现逻辑图

2.5.2 STM32与上位机通信

ROS和STM32控制器之间通过串口实现通信，通信协议包括：STM32控制器向ROS发送数据，ROS向STM32控制器发送数据。STM32向ROS发送数据使用“data_task”的任务。发送的数据帧包括三轴速度、角速度、加速度以及通信协议包含的帧头帧尾、校验位等信息。STM32接收数据采用中断接收的方式，接收的数据包括使能控制标志位、机器人三轴目标速度、数据校验位。

ROS运行在Jetson Nano上面，STM32采集底盘IMU、编码器的信息，然后通过串口3和Jetson Nano进行传感器数据的传输和控制指令的发送。ROS向STM32发送目标三轴的运动速度和云台舵机角度，以控制机器人运动：首先创建keyboard_teleop节点，发布三轴速度控制命令相关话题，然后wheeltec_robot节点订阅，通过串口通信发送到STM32。

2.5.3 nano和PC的通信

Secure Shell（SSH）是一种建立在应用层基础上的安全网络协议，可以用来远程登陆。本文使用的是SSH

在Ubuntu中的免费开源实现，使用它可以方便远程登录Jetson nano并操作机器人，如ROS系统发送的三轴的目标速度可以由PC端的键盘输入。

三、机器人功能实现及分析

搭建的步兵机器人样机实物如图4.1所示。

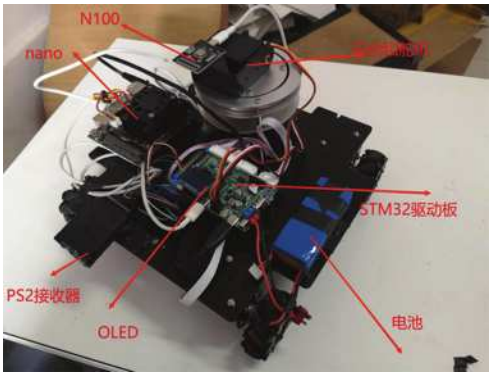

图4.1　机器人实物图

3.1 步兵机器人功能测试

3.1.1 nano和PC的通信测试

PC端的Ubuntu系统和Jetson nano通过SSH远程登陆通信。SSH提供两种级别的验证方法：第一种级别（基于口令的安全验证）：使用帐号和口令登录到远程主机。所有传输的数据都会被加密。口令登录命令格式为：ssh 客户端用户名-Y@服务器IP地址。输入上述命令，命令提示符会修改为远程主机的提示符，证明执行完命令之后操作的是Jetson nano。

3.1.2 人机交互与运动控制测试

OLED上显示四轮的目标速度与实际速度，根据实际需要可修改为显示Z轴角速度或底盘IMU模块（ICM20948）解算出的yaw航向角。

底盘的全向移动和云台的转动可以由PS2或者PC端控制完成。PC端通过上位机发送的控制命令优先级高于PS2（STM32串口3中断接收）。图4.2是PC控制底盘运动的显示界面，在登录后的的终端上进行，云台控制同理。

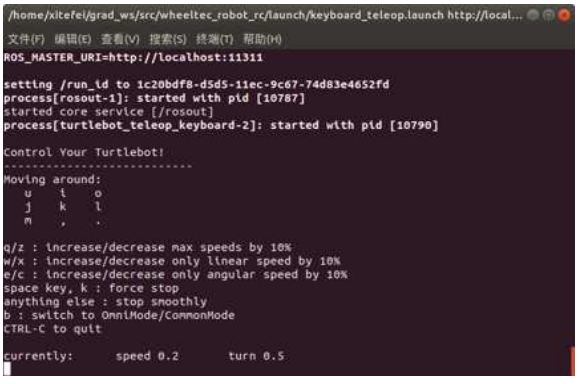

图4.2　PC控制显示界面

PC端通过ROS在命令行终端里发布的部分N100惯导模块参数如图4.3。

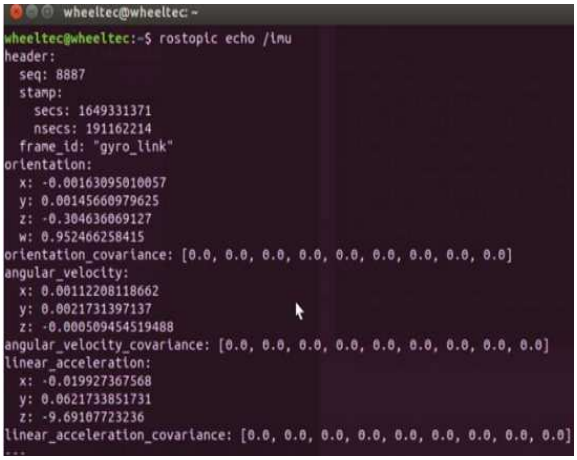

图4.3　PC终端显示IMU数据

3.2 小陀螺行进测试及姿态解算误差分析

测试结果表明替代方案可以基本实现小陀螺进行。但是长时间运行会产生数据漂移，导致机器人小陀螺行进轨迹漂移。

记录一段时间内IMU姿态解算的yaw轴航向角数据，以逆时针方向为正。如表4.1。

表4.1　PC终端显示IMU数据

模块	次数	测量角度（单位°）			
		0	90	180	270
N100	1	0.0	90.1	180.0	270.0
	2	0.1	90.1	180.2	269.9
ICM20948	1	0.0	89.2	180.4	269.9
	2	0.5	90.8	181.3	272.2
	3	3.4	94.4	183.5	275.5
	4	6.5	97.8	188.8	279.0
	5	10.7	101.1	193.6	284.9

显然，云台IMU N100的IM u数据满足设计要求。底盘IMU姿态数据存在明显的漂移误差。原因分析如下：

（1）龙格-库塔法（Runge-Kutta）解算yaw轴航向角本质上是对Z轴角速度及进行积分，虽然处理了零漂，但很难消除漂移的影响。

（2）九轴IMU解算yaw航向角只能用陀螺仪和磁力计，ICM20948集成在STM32驱动板上，和搭建的机器人各模块间距不大，磁力计容易受到影响且校准，因此引入ICM20948的三轴磁力计效果得不到保障。

在RoboMaster中，小陀螺的实现依靠云台yaw轴电机的绝对编码器和云台上的IMU模块。步兵机器人是允许配备导电滑环和全向电机的，因此本节小陀螺测试中

的这个问题可以避免。

四、结束语

本文进行了基于ROS系统的步兵机器人设计，搭建了机器人样机，使基于linux和ROS的Jetson nano和PC承担更多的任务。

目前ROS已经得到了广泛应用，开发者数量激增，设计领域越来越广。同时，在RM新的赛制中，以自动步兵为例，比赛中机器人自主决策无操作手，但可以接收雷达站给出的的基于小地图的两方相对位置。那么装在云台上，数据直接被机载PC接收的惯导模块就可以帮助机器人确定导航坐标系。并且接在机载PC上数据接收更及时，同时节省了STM32的计算资源。

当然，随着技术的进步和融合，更好的控制方案和更优秀的传感器肯定也会层出不穷。

参考文献：

[1]Marin P P, Hussein A, Martin D, et al. Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles[J]. Journal of Advanced Transportation, 2018.

[2]刘浩，陈恩庆，酒明远，等.基于ROS的环境感知小车设计与实现[J].现代电子技术，2021，44（12）：182-186.

[3]邵壮.兼容ROS的嵌入式实时机器人通信系统的设计与实现[D].哈尔滨工业大学，2018.