摘要

移动操纵机器人将底盘与机械臂进行整合，工作范围远远宽于固定机械臂，近年来在具身智能、仓储物流、工业装配等领域运用越发广泛。移动操纵机器人需要完成的任务往往不是单一步骤的，而是多个连续的有先后关系的操作组成的，对运动精度、感知稳定性、规划能力都有较高要求。现有工作存在许多明显的问题：大部分研究只专注某个子系统，而从底层驱动到上层决策的全栈研究较少；不同平台之间代码复用率低，更换机器人本体后就需要大量重构代码；机械臂的动力学模型在实际部署时精度大幅降低；底盘和机械臂的联合控制也没有得到足够重视。本文以 RoboMaster（RM）机甲大师赛为主要验证场景，针对上述问题研究多形态移动操纵机器人的系统设计和全身控制方法。

嵌入式部分将 STM32H723 双板作为核心计算单元，设计了五层分层软件框架，自顶向下为 HAL、BSP、驱动、中间件、应用，层间接口固定，宏定义用于平台配置。在本框架下，RM 多种构型的机器人，如全向轮步兵、麦克纳姆轮英雄、双 yaw 轴哨兵、7 自由度机械臂工程、轮腿等五种形态共用同一套框架代码，不同平台之间切换往往只需要在应用层进行更改。在通信层面，三条总线并行运行：FDCAN 负责电机控制，UART 负责裁判系统通信和视觉上位机通信，SPI 采 IMU 数据；DJI 和达妙两种电机的协议统一封装在驱动层，上层可以使用统一的力矩接口下发指令。超级电容管理用单独的 STM32F334 控制板，使用带有 HRTIM 的高精度 PWM 驱动双向 Buck-Boost 变换器，实现功率-电流-电压三环级联控制，超级电容对短路、过压、欠压、过温等机器人危险工况都进行了保护。云台使用用串级 PID 进行控制，此外使用了重力补偿与旋转前馈优化控制效果；云台姿态解算使用 Mahony 互补滤波与卡尔曼滤波，当检测到大幅振动时再使用一个贝塞尔预滤波器。

7 自由度机械臂控制为核心难点，单纯 PD 控制在全伸展位置无法抵消重力的影响，机械臂负载变化或关节摩擦增加控制效果便大幅下降。本文机械臂控制使用了三层结构：PD 反馈为底层基础，Pinocchio 结合 CppADCodeGen 从 URDF 自动生成 RNEA 逆动力学嵌入式 C 代码作为模型前馈，最上层添加在线补偿，把模型预测力矩和电机实测力矩的差值经卡尔曼滤波估计、低通平滑后进行叠加，对负载变化和摩擦等未在模型中考虑的因素持续修正。遥操作基于同构控制器，重力补偿、位置跟踪、力反馈三个模式可以在根据使用者的需求自由切换。轮腿平衡控制使用了增益调度 LQR 加 VMC Jacobian 力映射的方案，同时也实现了 NMPC 和跳跃状态机。感知这边，自动瞄准走 YOLO 加 EKF 的流水线；工程机器人的抓取任务用 GroundingDINO 和 FoundationPose

做零样本 6 自由度位姿估计，使用状态机进行 SEARCH 和 TRACK 两模式的切换，在 Jetson Orin 上进行了部署；自主导航集成了 Point-LIO SLAM、Nav2 和行为树。

根据实验结果，底盘阶跃响应时间小于 120 ms，云台跟踪误差小于 0.5°，功率利用率大于 95%，超级电容充放电效率达到 90% 以上，CAN 丢帧率低于 10^{-6}。在使用重力补偿的情况下，机械臂静态位置误差从 3°–8° 降低到 0.5° 以下；在此基础上使用在线补偿，动态跟踪误差将再下降 30%。轮腿机器人最高速度达到 2.0 m/s，具有跨越高度为 0.15 m 的台阶的能力。

关键词：移动操纵机器人，多形态平台，全身控制，机器人系统设计，具身智能

Research on System Design and Whole-Body Control Methods for Multi-Morphology Mobile Manipulators

David Hong (Computer Application Technology)
Directed by: Shanghang Zhang, Assistant Professor

ABSTRACT

Mobile manipulators combine a wheeled base with a robotic arm, making them well-suited for tasks that require both locomotion and dexterous manipulation—applications like factory assembly, warehouse logistics, and embodied intelligence scenarios come to mind. Getting all the subsystems to work together reliably is, in practice, quite difficult. Most published systems focus on one piece of the puzzle (the arm, or the chassis, or perception) rather than the whole stack. Code written for one robot morphology rarely carries over cleanly to another. Arm dynamics models that look reasonable on paper often perform poorly on real hardware once joint friction and cable compliance enter the picture. And coordinating the base and arm as a unified system is still not a solved problem. This thesis tackles these issues using RoboMaster competition robots as the experimental platform.

On the embedded side, two STM32H723 microcontrollers run a five-layer firmware stack organized as HAL, BSP, driver, middleware, and application. Keeping the layer interfaces fixed and using macro-based configuration meant the same codebase could be adapted to five robot types without rewriting from scratch: omni-wheel infantry, Mecanum-wheel hero, dual-Yaw-axis sentry, 7-DoF engineer arm, and the serial-leg wheel-legged platform. Communication runs over three buses in parallel—FDCAN for real-time motor control, UART for the referee system and teleoperation link, SPI for IMU data—with DJI and DaMiao motor protocols abstracted behind a common torque interface. A separate STM32F334 board handles supercapacitor energy management via a bidirectional Buck-Boost converter; the controller runs a three-loop cascade (power / current / voltage) with incremental PID and hardware protection for short-circuit, overvoltage, undervoltage, and overtemperature conditions. Gimbal attitude estimation uses a Bessel pre-filter feeding into either Mahony complementary filtering or a Kalman filter depending on vibration level, with gravity compensation and spin feedforward added to the cascade PID.

Getting the 7-DoF arm to hold position accurately turned out to be the central challenge. A plain PD controller works passably at low loads but cannot compensate gravity at full extension, and certainly cannot adapt to changing payloads or joint-level friction. The approach taken here layers a model feedforward term on top of the PD baseline. Pinocchio computes the RNEA inverse dynamics symbolically; CppADCodeGen then turns that computation into self-contained C code that runs directly on the STM32 without any external library dependency. On top of this, an online compensator tracks the gap between the model's predicted torque and what the motor actually reports, smooths it through a Kalman filter and a low-pass filter, and folds the result back into subsequent commands. This lets the controller adapt to load shifts and friction variations without needing an explicit model of either. Teleoperation is handled through an isomorphic bilateral controller that supports gravity-compensation, position-tracking, and force-feedback modes with in-field switching. For the wheel-legged robot, gain-scheduled LQR combined with VMC Jacobian force mapping handles most operating conditions; NMPC was also implemented for larger disturbances, along with a jump state machine.

On the perception side, armor-plate detection runs a YOLO detector with EKF-based state estimation for moving targets. For manipulation tasks, GroundingDINO and FoundationPose are combined for zero-shot 6-DoF pose estimation—no target-specific training required. A SEARCH/TRACK state machine switches between the two modes on the Jetson Orin. Autonomous navigation uses Point-LIO for lidar-inertial odometry, Nav2 for path planning, and a behavior tree for the sentry's patrol logic.

Hardware tests gave the following results. Chassis step response settled within 120 ms. Gimbal angle tracking error stayed below 0.5° under normal operation. Power utilization reached above 95%, and supercapacitor charge-discharge efficiency above 90%. CAN frame-loss rate was below 10^{-6}. Adding gravity compensation brought static arm position error down from a 3°–8° baseline to within 0.5°; the online compensator reduced residual dynamic tracking error by roughly 30% on top of that. The wheel-legged robot ran stably at up to 2.0 m/s and cleared 0.15 m step obstacles without falling.

KEY WORDS: Mobile Manipulation Robot, Multi-Morphology Platform, Whole-Body Control, Robot System Design, Embodied Intelligence