

ShineMinxing / Ros2Go2Estimator

<> Code Issues 1 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Use this template

☆ 95 stars ⚡ 8 forks 🏃 1 watching Branches Activity Tags

Public template repository

main ▾ 2 Branches 0 Tags Go to file Go to file Add file ...

ShineMinxing use kinematic and footfall position to correct orientation could work... 4fb284d · 3 weeks ago

fusion_estimator use kinematic and footfall position to correct ... 3 weeks ago

.gitmodules move others to Ros2Go2Base 4 months ago

Readme.md readme update and correct estiamtor reset 2 months ago

config.yaml use kinematic and footfall position to correct ... 3 weeks ago

README

Ros2Go2Estimator

License MIT

Ros2Go2Estimator 是四足/双足机器人在 ROS 2 Humble / Ubuntu 22.04 上的 估计算法仓库，主要负责使用 IMU + 关节编码器 + 足底压力，给出高精度里程计结果

✨ 估计算法特性

类别	说明
双足 / 四足通用	行进中自动识别支撑足，无需切模式；支持站立 / 行走快速切换
全 3D & 平面 2D	同时发布完整 6DoF 里程计 (SMX/Odom) 与压平 Z-轴的 2D 里程计 (SMX/Odom_2D)
零依赖外部传感器	无视觉、激光传感器也能获得 <1 % 累积误差 ¹
运行时参数调节	三轴补偿角等参数可通过 ros2 param 在线微调

🏗 生态仓库一览

范畴	仓库	功能简介
底层驱动	https://github.com/ShineMinxing/Ros2Go2Base	DDS 桥、Unitree SDK2 控制、点云→Scan、TF
里程计	Ros2Go2Estimator (本仓库)	纯运动学多传感器融合
语音 / LLM	https://github.com/ShineMinxing/Ros2Chat	离线 ASR + OpenAI Chat + 语音合成
图像处理	https://github.com/ShineMinxing/Ros2ImageProcess	相机、光点/人脸/无人机检测
吊舱跟随	https://github.com/ShineMinxing/Ros2AmovG1	Amov G1 吊舱控制、目标跟踪

范畴	仓库	功能简介
工具集	https://github.com/ShineMinxing/Ros2Tools	蓝牙 IMU、手柄映射、吊舱闭环、数据采集

⚠ 按需克隆：若只想提升状态估计，可仅使用本仓库。其它仓库互不强依赖。

📁 本仓库结构

```
Ros2Go2Estimator/
├── fusion_estimator/
│   ├── launch/          # 源码包 (ROS2 节点)
│   │   ├── launch        # 示例 launch 文件
│   │   └── urdf/         # 机器人 URDF
│   ├── cfg/              # ROS2相关接口
│   └── src/
│       ├── fusion_estimator_node.cpp # ROS2相关接口
│       └── Go2FusionEstimator      # 纯C++实现，可移植到ROS1
└── config.yaml          # 全局参数 (见下)
    └── Readme.md          # ← 你正在看
```

⚙ 参数一览 config.yaml

参数	默认值	说明
sub_imu_topic	SMX/Go2IMU	订阅 IMU
sub_joint_topic	SMX/Go2Joint	订阅关节状态
sub_mode_topic	SMX/SportCmd	接收复位 / 模式切换指令 (可选)
pub_estimation_topic	SMX/Estimation	发布内部融合状态 (debug)
pub_odom_topic	SMX/Odom	发布 6DoF 里程计
pub_odom2d_topic	SMX/Odom_2D	发布 2D 里程计
odom_frame	odom	TF world frame
base_frame	base_link	机器人质心 Frame
base_frame_2d	base_link_2D	压平后的 2D Frame
urdf_file	cfg/go2_description.urdf	机器人描述文件，用于腿长 / 连杆参数
Modify_Par_[1-3]	0.0	运行时可调三轴补偿角 (°)

⚙ 安装与编译

```
# 1. clone 到工作空间
cd ~/ros2_ws/LeggedRobot/src
git clone --recursive https://github.com/ShineMinxing/Ros2Go2Estimator.git

# 2. 编译
cd .. && colcon build --packages-select fusion_estimator
source install/setup.bash

# 3. 运行 (示例)
ros2 run fusion_estimator fusion_estimator_node
```

📝 节点接口

```
/fusion_estimator_node (rclcpp)
|— 发布
```

```

|   • SMX/Odom      nav_msgs/Odometry (frame: odom → base_link)
|   • SMX/Odom_2D    nav_msgs/Odometry (frame: odom → base_link_2D)
|   • SMX/Estimation custom / debug
└ 订阅
  |   • SMX/Go2IMU    sensor_msgs/Imu
  |   • SMX/Go2Joint   sensor_msgs/JointState
  |   • SMX/SportCmd   std_msgs/Float64MultiArray (复位/模式)
└ 依赖 TF 发布由 message_handle 包完成

```



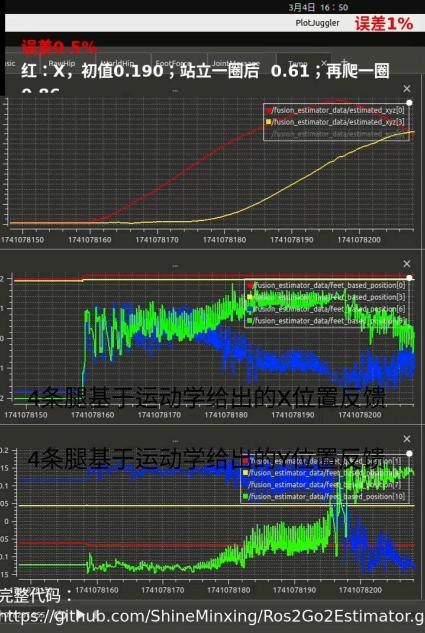
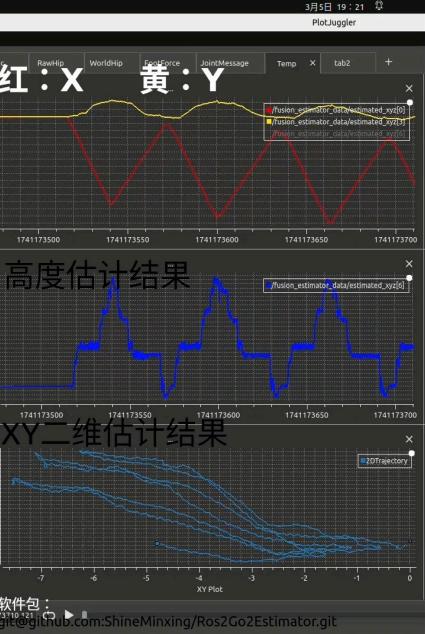
算法概览

1. 支撑足检测：根据足底力矩与接触逻辑动态识别支撑点。
2. 前向运动学：利用 URDF 中的连杆长度 + 关节角计算足端相对位姿。
3. 零速更新 (ZU)：支撑足静止期对 IMU 积分误差进行校正。
4. 基于四元数微分的姿态积分：避免欧拉角奇异。
5. 卡尔曼 / QP 融合：组合 IMU、足底里程对质心位置进行批量最小二乘纠正。
6. 2D 投影：将 roll/pitch 置零，提供 IMU-free 的平面 Odom。

详细推导请见技术白皮书 [Notion 文档](#)

视频演示

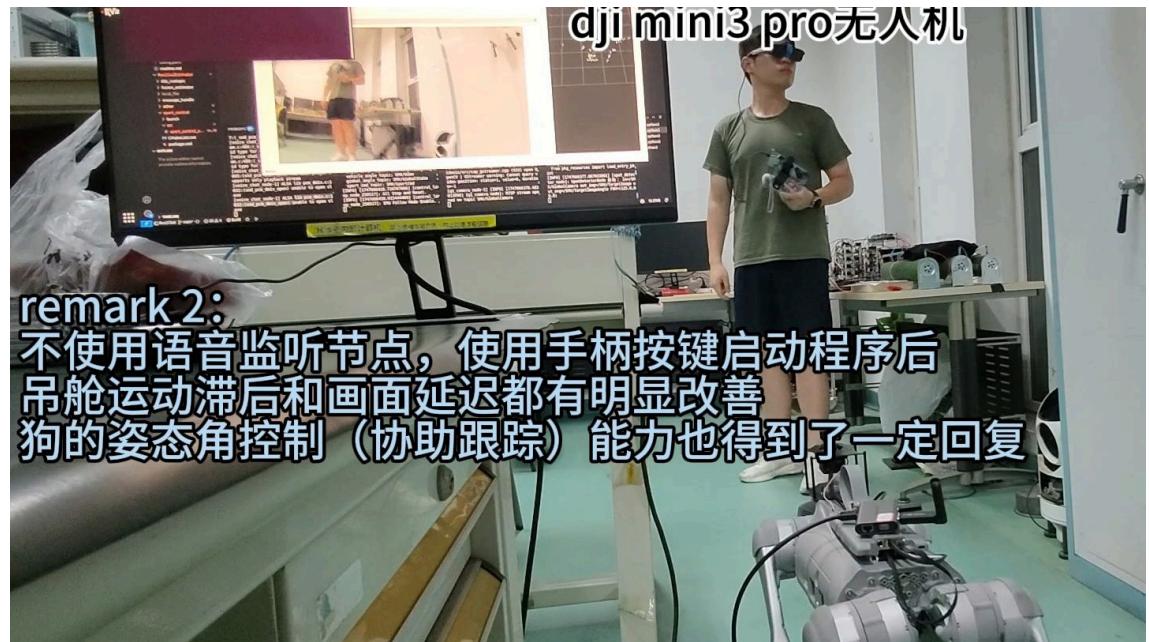
主题	点击图片观看
纯里程计建图 (站立/四足切换)	

主题	点击图片观看
<p>室内行走误差 0.5 %-1 %</p> 	 <p>完整代码： https://github.com/ShineMinxing/Ros2Go2Estimator.git</p>
<p>爬楼梯高度误差 <5 cm</p> 	 <p>软件包： git@github.com:ShineMinxing/Ros2Go2Estimator.git</p>

主题	点击图片观看
户外行走380m 误差 3.3 %	
语音交互 + 地图导航	<p style="color: red; font-size: 1.5em; margin-top: 10px;"> Go2机器人+腿式里程计+机器狗自带雷达SLAM预建地图+本地语音唤醒语音识别+DeepseekV3大模型API交互+阿里/本地文字转语音+Nav2导航 </p> <p style="font-size: 0.8em; color: red; margin-top: 10px;"> https://github.com/ShineMinxing/Ros2Go2Estimator.git </p>

主题

点击图片观看





Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● C++ 98.8% ● C 1.2%

Suggested workflows

Based on your tech stack

 C/C++ with Make	Configure
Build and test a C/C++ project using Make.	
 SLSA Generic generator	Configure
Generate SLSA3 provenance for your existing release workflows	
 CMake based, single-platform projects	Configure
Build and test a CMake based project on a single-platform.	

[More workflows](#)

[Dismiss suggestions](#)