Autonomous Tracked Agricultural UGV Configuration and Navigation Experimental Results

Authors
Flavio Callegati, Alessandro Samorì, Roberto Tazzari, Nicola Mimmo and Lorenzo Marconi
Precision Agriculture

Manual workers and Mobile Robots Co-operation
Main Topics & Purposes

- Introduction
- Hardware Structure
- Software Architecture
- Experiment Purposes
- Results
- Conclusions

To achieve Robustness:
- Mechanical Structure

To obtain Local Positioning:
- Sensor Suite
- Software Structure

Experimental Tests to evaluate performances
Hardware Structure

Mechanical Structure
Tracked Vehicle

PROs
• Robustness to uneveness
• Higher traction

CONs
• Skidding effect

Sensor Suite

GPS Receiver
- Global Positioning

3D Laser Scanner
- Local Positioning
- Obstacle Detection

Inertial Measurement Unit
- Attitude Estimation
Software Subsystem

Running on the onboard computer
- Intel NUC mini PC

Rover management and control
- User *interfacing*
- Sensor reading
- Data processing
- Motor *references* computation

Two basic software elements
- **HMI** (*High* level)
- **Control** system (*Low* level)
Human Machine Interface

Functions

- **Commands** assignment
- Mission **monitoring**

Appearence

- Google-like satellite **map**
- Buttons and labels

Bidirectional UDP communication with the Control system
Control System

Implemented in **ROS** (Robot Operating System)
- Executables (*ROS nodes*)

**Pose estimation (PE)**
- *Sensor* reading
- Position and attitude *estimation*

**Row State Machine (RSM)**
- User *commands* interpretation
- Trajectory computation
- Elementary functions *management*

**Row control**
- Feedback control
- Autonomous navigation
Control System

Kinematic model (Differential Drive)

\[ v: \text{translation velocity} \]
\[ \cdot \text{Depends on the task} \]
\[ \cdot \text{Usually small and constant} \]

\[ \omega: \text{rotation velocity} \]
\[ \cdot \text{ Defines the lateral distance} \]
\[ \cdot \text{Choosen as control variable} \]

Estimation of the lines of trees

\[ \cdot \text{Structured environment (Rows)} \]
\[ \cdot \text{Need of the relative position (lateral distance)} \]

Row navigation control

\[ \omega = \frac{1}{v \cos \theta} \left( \dot{d}_R - k_d (\dot{d} - \dot{d}_R) - k_p (d - d_R) \right) \]

\[ d_R : \text{lateral distance} \]
1. Recognition of being inside the rows and starting navigation

2. Keeping desired lateral distance

3. Detection of the row exit and stop navigation.
Results

• Good communication between the software layers

• No errors in the mission assignment

• Control law validated
Results

• Good estimation of row lines with *Hough’s* algorithm

• Estimation of distances and angles with respect to the rows

• Precision (polar coordinates):
  \[ \Delta d = 0.1 \text{m} \]
  \[ \Delta \vartheta = 2^\circ \]
Conclusions

• Software structure is flexible and modular

• Estimation of distance and heading localizes rover into rows.

• Merge this estimation with odometry, IMU and GPS data in order to improve rover localization and robustness
THANK YOU
FOR
YOUR ATTENTION