Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Software Development Project

Basic Navigation Test Analysis

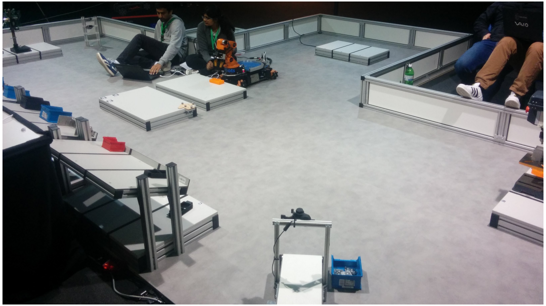
Wei-Chan Hsu, Torsten Jandt, Ramesh Kumar, Danning Wang

May 21, 2017

Basic Navigation Tasks

Scenario Environment

- Walls, service areas, places, arena objects
- Wall markers and floor markers
- Obstacles



Basic Navigation Tasks

Tasks

- Task Specification
  - Place
  - Orientation
  - Pause duration
- Purpose
  - Move to the places orderly
  - Finally leave the arena through the gate
- Requirement
  - Orient itself according to the orientation
  - Cover a place marker
  - Pause for the time in seconds

Navigation Subtasks and Challenges

Perception

- Reading and processing sensor data
- Challenges
    - Sensor error
    - Sensor calibration
    - Sensor fusion
    - Interpreting markers

Navigation Subtasks and Challenges

Mapping

- Building the map of environment
- Challenges
    - Measurement noise
    - Dynamic environment
    - High dimension
    - Barrier tape

Navigation Subtasks and Challenges

Localization

- Calculate current position in the map
- Challenges
    - Dynamic environment
    - Motion errors
    - Similar surroundings

Navigation Subtasks and Challenges

Path planning

- Determine a sequence of actions from start to goal position.
- Given: Map, start pose, goal pose, robot geometry
- Map: representation, update
- Types: Grid-based, Graph-based, Field-based
- Search algorithms
- Challenges: Accessibility problem, computational complexity, uncertainties, dynamic environment

Navigation Subtasks and Challenges

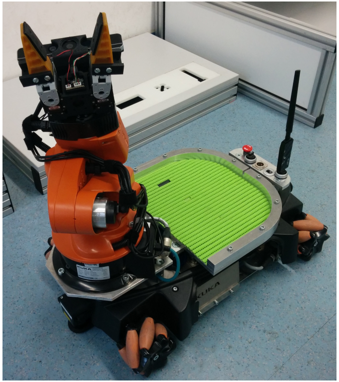
Motion control

- Subtasks: Path execution and acting
- Move robot from point to point or follows a trajectory
- Input: Path commands, output: motor commands to wheels
- Odometry feedback
- Requirements: Kinematic model
- Robot kinematics influences the maneuverability and controllability.
- Challenges: System modeling complexity, system errors

KUKA youBot

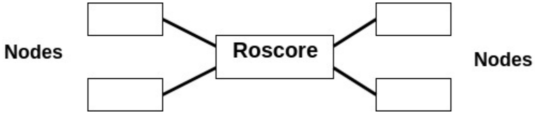
The youBot is a mobile manipulator designed for education and research purposes. It comes with fully open interfaces and API.

- Omnidirectional, four-wheeled
- 5-DOF manipulator with a two-finger gripper
- On-board PC with CPU, 2GB memory, 32GB SSD drive
- Sensors: vision sensors, rangefinders

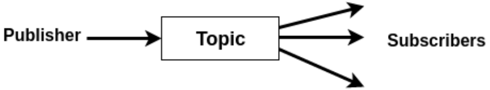


Robot Operating System (ROS)

- The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.
- Node: A process that uses ROS to communicate with other nodes.

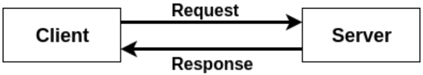


- Topic: Mechanism to send messages from a node to one or more nodes.
- Topic follows a publisher-subscriber design pattern.



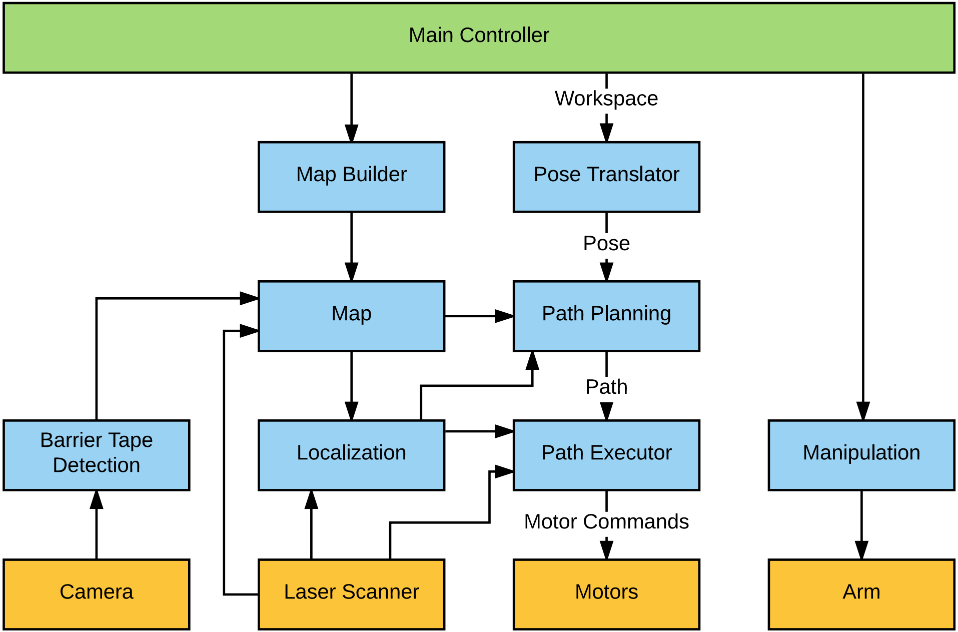
continue...

- Service: Mechanism for a node to send a request to another node and receive a response in return.
- Service follows a request-response design pattern.



- ROS supports two programming languages:
  - C++
  - Python

Software Modules



Complexity

System

- High complexity, but easy to maintain
- Split into several modules

Modules

- Represent ROS nodes
- Easy replaceable

Interfaces

- Specified by topic names and message types
- Modules subscribe to topics as needed

Workplan

1. Getting to know the youBot
2. Investigate state of the art
    - What is already implemented by ROS
    - What can be used
    - What needs to be implemented
3. Create modules
4. Integrate modules
5. Test system
6. Extend solution
    - Barrier tape detection