Mobile Target Tracking Using Radio Sensor Network

Nic Auth   Grant Hovey   Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Saturday, April 28, 2018

Outline

Motivation

- Mobile robotics is an area of active research
- Mobile robotics is an interdisciplinary field with many applications
- No literature on active RF Mobile Target Tracking

Problem Description

A mobile target is moving randomly in an unknown environment where its position and trajectory are also unknown. A mobile robot is to follow the mobile target while maintaining a set distance and avoiding obstacles in its path.

Problem Statement

- Develop a modular mobile target tracking strategy using RF signal strength measurements
- Build a simulation to test the EKF-SLAM with mobile target tracking algorithm
- Implement the mobile target tracking strategy using ROS on the Pioneer 3-DX mobile robot platform

- Warehouses
- Factories
- Search and rescue
- Research and education

- Overcoming noisy received signal strength measurements
- Localization of moving target
- Speed of radio sensor signal acquisition

- **EKF**: *Extended Kalman Filter*
- **SLAM**: *Simultaneous Localization and Mapping*
- **RSSI**: *Received Signal Strength Indicator*
- **ROS**: *Robot Operating System*

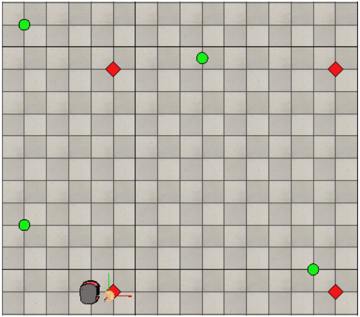
Overview

Background Study

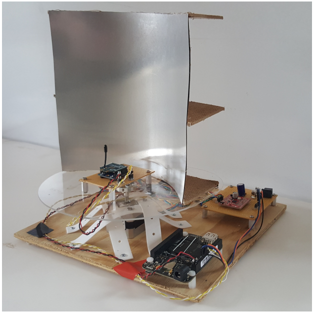
Literature

- Radio signal strength to calculate range
- Parabolic reflector to estimate angle of arrival
- Simultaneous localization and mapping using RFID tags
- Tracking of radio signal emitting mobile target

- Radio receiver and robot control hardware
- Radio sensor network
- EKF-SLAM Simulation and Implementation

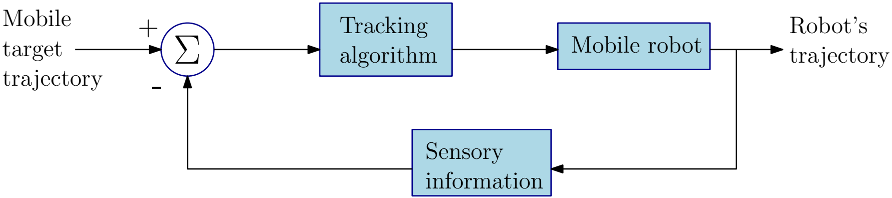


(a) Operating environment in simulation.



(b) Customized radio transceiver.

Overview



Mobile target trajectory
+
-
$\sum$
Tracking algorithm
Mobile robot
Robot's trajectory
Sensory information

EKF: *Extended Kalman Filter*
SLAM: *Simultaneous Localization and Mapping*

- Widely popular for resilience to noise.
- Three main steps:
  1. Predict
  2. Observe
  3. Update

Predict Step

- Predict the state of the robot, the mobile target, and the measurements to be made

$$
\begin{aligned}
\hat{\mathbf{q}}_{k+1}^{-} &= \hat{\mathbf{q}}_{k}^{+} + \mathbf{C}^{T}\mathbf{f}_{r}(\hat{\mathbf{q}}_{k,r}^{+}, \mathbf{u}_{k}, \mathbf{0}), && \text{(1a)} \\
\mathbf{P}_{k+1,rr}^{-} &= \mathbf{F}_{k}\mathbf{P}_{k,rr}^{+}\mathbf{F}_{k}^{T} + \mathbf{L}_{k}\mathbf{Q}_{k}\mathbf{L}_{k}^{T}, && \text{(1b)} \\
\mathbf{P}_{k+1,r\mathcal{B}}^{-} &= \mathbf{F}_{k}\mathbf{P}_{k,r\mathcal{B}}^{+}, && \text{(1c)} \\
\mathbf{P}_{k+1,\mathcal{B}r}^{-} &= \left(\mathbf{P}_{k+1,r\mathcal{B}}^{-}\right)^{T} && \text{(1d)}
\end{aligned}
$$

Observations

- Take range and bearing measurements of the radio sensor network

$$Y_t = [r_t, \theta_t]^T \quad (2)$$

Update Step

- Update the predictions based on the measurements gathered
- Update the Kalman gain covariances

$$\mathbf{S} = \begin{bmatrix} \mathbf{H}_r & \mathbf{H}_{\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{-}_{k+1,rr} & \mathbf{P}^{-}_{k+1,r\mathbf{b}^j} \\ \mathbf{P}^{-}_{k+1,\mathbf{b}^j r} & \mathbf{P}^{-}_{k+1,\mathbf{b}^j\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{H}_r^T \\ \mathbf{H}_{\mathbf{b}^j}^T \end{bmatrix} + \mathbf{R}, \tag{3a}$$

$$\mathbf{K}_{k+1} = \begin{bmatrix} \mathbf{P}^{-}_{k+1,rr} & \mathbf{P}^{-}_{k+1,r\mathbf{b}^j} \\ \mathbf{P}^{-}_{k+1,\mathcal{B}r} & \mathbf{P}^{-}_{k+1,\mathcal{B}\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{H}_r^T \\ \mathbf{H}_{\mathbf{b}^j}^T \end{bmatrix} \mathbf{S}^{-1}, \tag{3b}$$

$$\hat{\mathbf{q}}^{+}_{k+1} = \hat{\mathbf{q}}^{-}_{k+1} + \mathbf{K}_{k+1}\mathbf{v}, \tag{3c}$$

$$\mathbf{P}^{+}_{k+1} = \mathbf{P}^{-}_{k+1} - \mathbf{K}_{k+1}\mathbf{S}\mathbf{K}^T_{k+1}, \tag{3d}$$

Augment

- Add any new landmarks gathered
- Increase the size of the prediction covariance so more covariance matricies can be stored

$$q_{k+1} = \begin{bmatrix} q_{k+1}^{+} \\ b^{s+1} \end{bmatrix}$$

$$P_{k+1} = \begin{bmatrix} P_{k+1}^{+} & P_{b^{(s+1)}q}^{T} \\ P_{b^{(s+1)}q} & P_{b^{(s+1)}b^{(s+1)}} \end{bmatrix}$$

Discrete time model:

$$x_{k+1} = x_k + T\nu_k \cos(\theta_k + \gamma_k),$$
$$y_{k+1} = y_k + T\nu_k \sin(\theta_k + \gamma_k),$$
$$\theta_{k+1} = \theta_k + T\nu_k \frac{\sin(\gamma_k)}{l}$$

- Model of RSSI $z_k^{[j]}$

$$z_k^{[j]} \approx P_{ref} - 10\eta \log_{10} r_k^{[j]} \tag{4}$$

- $z_k^{[j]}$ measured in dBm, P_{ref} is the reference signal strength in dBm at 1m, η is the signal propagation constant, $r_k^{[j]}$ is the range in m between the robot and the $[j]^{th}$ radio sensor
- Bearing β_k of each radio sensor is determined from a set of RSSI measurements

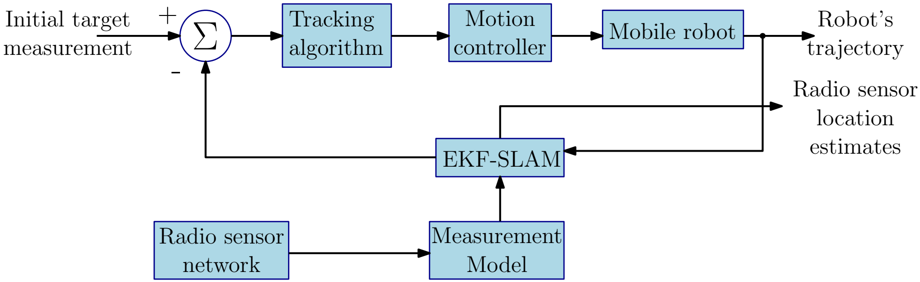
- Range is calculated using the RSSI

$$r_k = 10^{\frac{|\mathsf{RSSI}| - |P_{\mathsf{ref}}|}{10\eta}} \tag{5}$$

- RSSI is assumed to be greatest when the parabolic dish is directly facing the Xbee. This angle is determined to be the bearing β_k.

Mapping and Tracking Methods

Subsystem Level Diagram



Overview

BRADLEY University

Simulation

Virtual Robot Experimentation Platform

- Commercial robotics simulator
- Accurate models of many robots including the Pioneer 3-DX

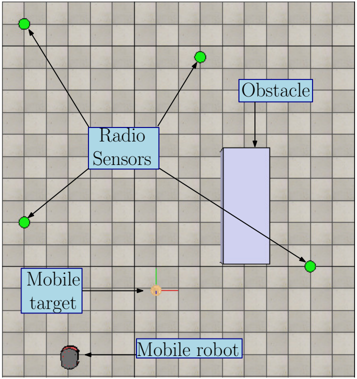
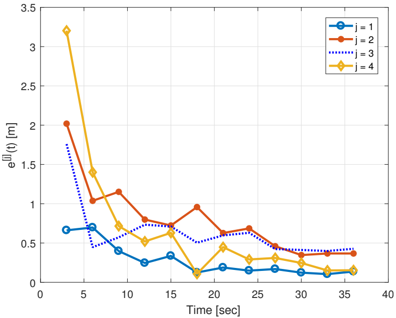


Figure: Virtual environment

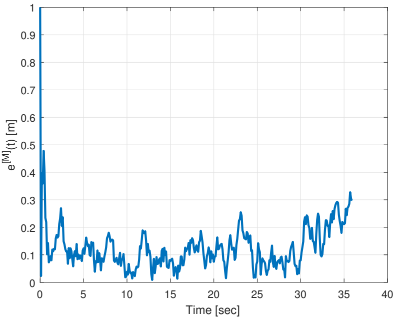
- **Purpose:** To verify the EKF-SLAM and Mobile Target Tracking algorithms before implementing

Play video of V-Rep and Matlab work

Simulation Case I: Results

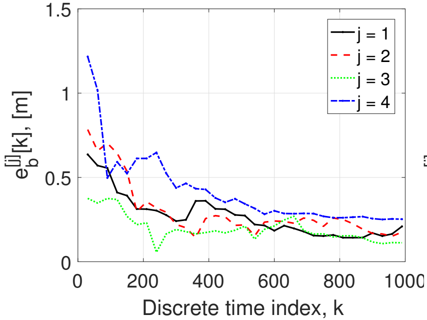


(a) Radio sensor estimation errors

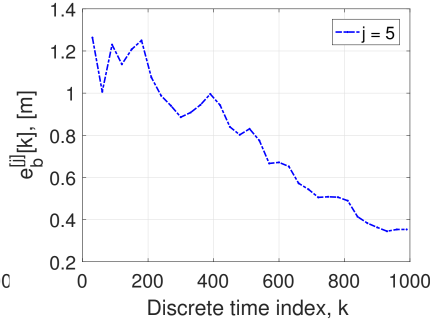


(b) Mobile target estimation error

Play video of Matlab work



(a) Radio sensor estimation errors



(b) Mobile target estimation error

Overview

BRADLEY University

- Full featured operating system
- Easily modifiable and open source
- Version adapted for beaglebone black

Implementation
Software

- Collection of robotics middle-ware
- Hardware abstraction
- Node based system
- TCP/IP based communication

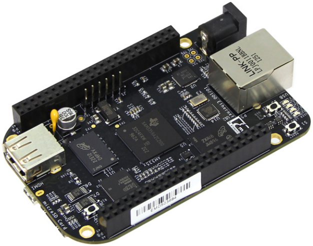
- MATLAB
  1. Feature rich scripting language
  2. ROS support with Robotics System Toolbox
  3. Used to process data and perform SLAM algorithm
- C++
  1. Fast yet high level programming language perfect for embedded applications
  2. CMake build tools included with ROS
  3. Used to operate customized radio transceiver

Implementation

Hardware

- Zigbee Radio Protocol
- 2.4 GHz Frequency
- 10 to 20 meter range
- Low power consumption

Implementation
Hardware



- 1 GHz Cortex A-8 ARM Processor
- UART Interface
- Wifi Capable
- Low Cost - $55.00

Implementation
Hardware

- Purpose built for research
- 500 bit wheel encoders
- 8 on board sonar sensors
- ARIA framework

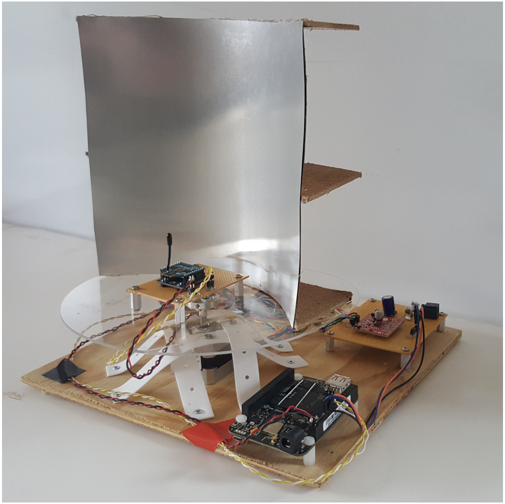
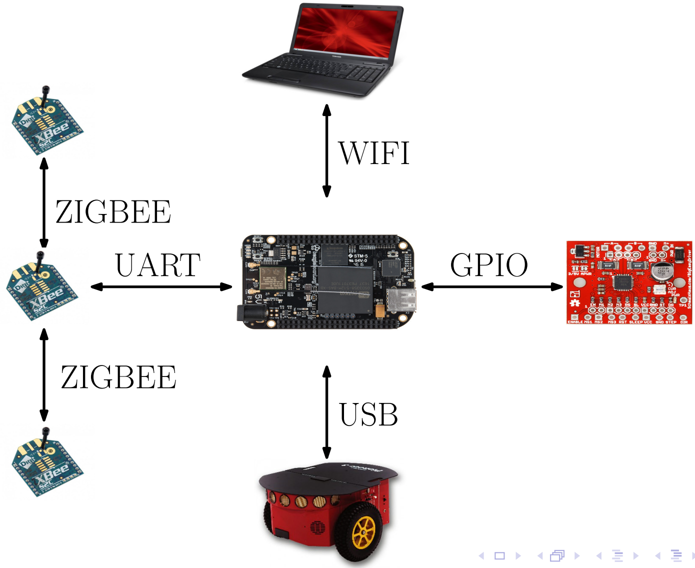
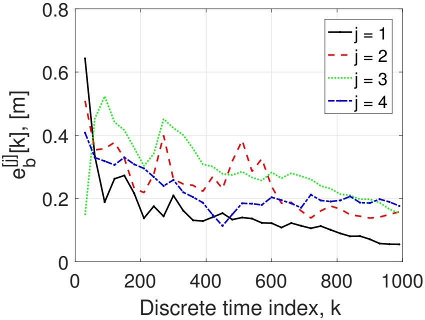


Figure: Customized radio transceiver

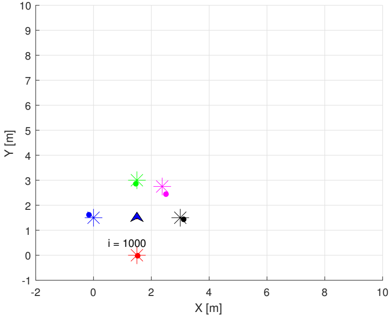
Implementation
Communication Diagram



Overview



(a) Radio sensor estimation errors



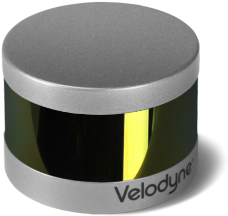
(b) Testing environment

Play video of implementation

Overview

Future Directions

- Improve speed of radio sensor signal acquisition
- Reduce error due to multipath propagation
- Potentially incorporate other sensor types such as LIDAR



Conclusion

- Excellent performance in simulation
- Adequate performance in actual implementation
- Room for improvement or expansion of scope

For Further Reading I

- F. Martinelli.
  A robot localization system combining rssi and phase shift in...
  IEEE Transactions on Control Systems Technology, vol.23, no. 5, pp. 1782–1796.
- E. DiGampaolo.
  A robot localization system using the phase of passive uhf-rfid...
  IEEE Transactions on Control Systems Technology, vol.61, no. 1, pp. 365–376.
- E. Xu.
  Target tracking and mobile sensor navigation in wireless sensor...
  IEEE Transactions on Mobile Computing, vol.12, no. 1, pp. 177–186.
- www.mobilerobots.com/Libraries/Downloads/Pioneer3DX-P3DX-RevA.sflb.ash