Indoor Robot Localization and Mapping Using ZigBee Radio Technology

Kyle Hevrdejs Jacob Knoll
Advisor: Dr. Suruz Miah

Senior Capstone Project Presentation, 2017
Electrical and Computer Engineering Department
Bradley University

May 2, 2017

Outline

1. Introduction
2. Navigation and Mapping
   - EKF-SLAM Algorithm
   - Measurement Model
   - Motion Control Strategy
3. V-REP Simulation
   - Background and Setup
   - Simulation Cases
4. Customized Radio Transceiver
   - Range and Bearing Approximation
   - Hardware and Software
   - Initial Experimental Results
5. Implementation
   - Experimental Cases
6. Future Directions and Conclusion

BRADLEY University

Introduction

- Simultaneous localization and mapping (SLAM) is a popular research topic
- Many papers only address localization, mapping, or navigation individually
- Solutions are often too expensive

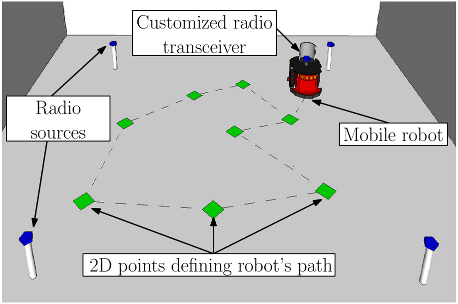


Figure: Illustration of the experimental setup

Introduction

Motivation

- Most localization and mapping implementations are expensive
- Implemented systems have target platforms
- No literature uses range-only measurements

Challenges

- Using an easily adaptable model
- Overcoming the noise in RSSI measurements
- Determining the Angle-of-Arrival (AoA) of signals
- Create a cost–effective solution

Objective

- Implement a transceiver capable of using range-only measurements
- Simulate Extended Kalman Filter Simultaneous Localization and Mapping (EKF-SLAM) in a commercial robot simulator
- Implement the EKF-SLAM algorithm using ROS on the Pioneer 3-DX

Currently, the work we have completed has been accepted for publication in two international conferences:

1. The 26th IEEE International Symposium on Industrial Electronics, 19-21 June 2017 in Edinburgh, Scotland, UK
2. The 30th Annual IEEE Canadian Conference on Electrical and Computer Engineering

- **EKF**: Extended Kalman Filter
- **SLAM**: Simultaneous Localization and Mapping
- **LoS**: Line-of-Sight
- **RSSI**: Received Signal Strength Indicator
- **WPAN**: Wireless Personal Area Network
- **RMSE**: Root Mean Square Error
- **ROS**: Robot Operating System

Outline

- Popular in the robotics community for its resilience to noise
- Multiple steps
  1. Predict and measure
  2. Update
  3. Augment

Predict

- Predict the state of the robot and the measurements that will be made

$$\hat{\mathbf{q}}_{k+1}^{-} = \hat{\mathbf{q}}_{k}^{+} + \mathbf{C}^{T}\mathbf{f}_{r}(\hat{\mathbf{q}}_{k,r}^{+}, \mathbf{u}_{k}, \mathbf{0}) \tag{1a}$$

$$\mathbf{P}_{k+1,rr}^{-} = \mathbf{F}_{k}\mathbf{P}_{k,rr}^{+}\mathbf{F}_{k}^{T} + \mathbf{L}_{k}\mathbf{Q}_{k}\mathbf{L}_{k}^{T} \tag{1b}$$

$$\mathbf{P}_{k+1,r\mathcal{B}}^{-} = \mathbf{F}_{k}\mathbf{P}_{k,r\mathcal{B}}^{+}, \tag{1c}$$

$$\mathbf{P}_{k+1,\mathcal{B}r}^{-} = \left(\mathbf{P}_{k+1,r\mathcal{B}}^{-}\right)^{T} \tag{1d}$$

Measure

- Take measurements for use in the update step
- Detailed in a later slide

Update

- Update the predictions based on measurements
- Update covariances

$$\mathbf{S} = \begin{bmatrix} \mathbf{H}_r & \mathbf{H}_{\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{P}^-_{k+1,rr} & \mathbf{P}^-_{k+1,r\mathbf{b}^j} \\ \mathbf{P}^-_{k+1,\mathbf{b}^j r} & \mathbf{P}^-_{k+1,\mathbf{b}^j\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{H}_r^T \\ \mathbf{H}_{\mathbf{b}^j}^T \end{bmatrix} + \mathbf{R}, \tag{2a}$$

$$\mathbf{K}_{k+1} = \begin{bmatrix} \mathbf{P}^-_{k+1,rr} & \mathbf{P}^-_{k+1,r\mathbf{b}^j} \\ \mathbf{P}^-_{k+1,\mathcal{B}r} & \mathbf{P}^-_{k+1,\mathcal{B}\mathbf{b}^j} \end{bmatrix} \begin{bmatrix} \mathbf{H}_r^T \\ \mathbf{H}_{\mathbf{b}^j}^T \end{bmatrix} \mathbf{S}^{-1} \tag{2b}$$

$$\hat{\mathbf{q}}^+_{k+1} = \hat{\mathbf{q}}^-_{k+1} + \mathbf{K}_{k+1}\mathbf{v} \tag{2c}$$

$$\mathbf{P}^+_{k+1} = \mathbf{P}^-_{k+1} - \mathbf{K}_{k+1}\mathbf{S}\mathbf{K}^T_{k+1} \tag{2d}$$

Augment

- Add any new features to the map
- Increase size of covariance matrix so new covariances can be stored

$$\mathbf{q}_{k+1} = \begin{bmatrix} \mathbf{q}^{+}_{k+1} \\ \mathbf{b}^{[s+1]} \end{bmatrix} \text{ and } \mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{P}^{+}_{k+1} & \mathbf{P}^{T}_{\mathbf{b}^{[s+1]}\mathbf{q}} \\ \mathbf{P}_{\mathbf{b}^{[s+1]}\mathbf{q}} & \mathbf{P}_{\mathbf{b}^{[s+1]}\mathbf{b}^{[s+1]}} \end{bmatrix}$$

- Ackermann model is used for modularity
- The discrete time model is shown in (3)

$$x_{k+1} = x_k + T\nu_k \cos(\theta_k + \gamma_k), \tag{3a}$$

$$y_{k+1} = y_k + T\nu_k \sin(\theta_k + \gamma_k), \tag{3b}$$

$$\theta_{k+1} = \theta_k + T\nu_k \frac{\sin(\gamma_k)}{\ell}, \tag{3c}$$

BRADLEY University

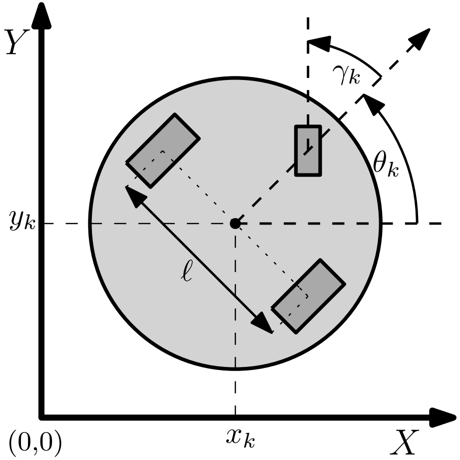


Figure: An example of the Ackermann steering vehicle

Outline

BRADLEY University

- Equation (4) shows the conversion of RSSI $z_k^{[j]}$ to range $r_k^{[j]}$

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

- The bearing $\beta_k^{[j]}$ of each XBee is determined from a set of RSSI measurements $\mathcal{R}^{[j]}$ using (5)

$$\beta_k^{[j]} = \underset{[-\pi,\pi)}{\arg\max}\, \mathcal{R}^{[j]} \tag{5}$$

- Measurement model of noisy LoS range and bearing where $\boldsymbol{\xi}_k \in \mathbb{R}^2$ is the noise (6)

$$\mathbf{y}_k^{[j]} = \left[r_k^{[j]}, \beta_k^{[j]}\right]^T + \boldsymbol{\xi}_k = \mathbf{h}(\mathbf{q}_{k,r}, \mathbf{b}^{[j]}, \boldsymbol{\xi}_k) \tag{6}$$

Outline

BRADLEY University

- Proportional (P) Controller
  - Tune the steering of the mobile robot
  - Easy to customize the response
- Fuzzy Logic Controller
  - Control linear velocity of mobile robot
  - Can model highly nonlinear systems if different responses are desired

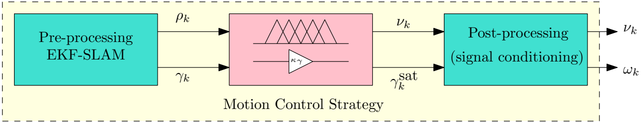


Figure: The implemented motion control strategy

Subsystem Block Diagram

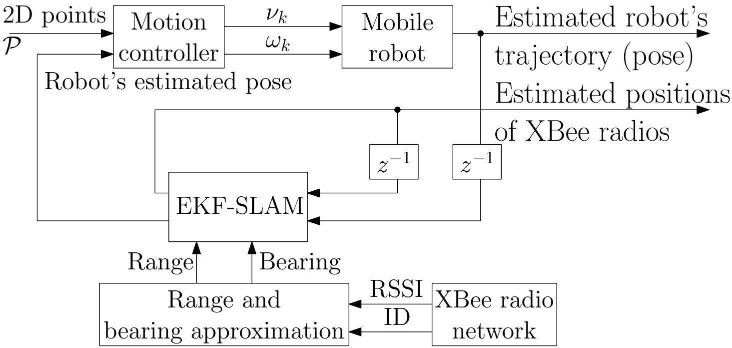


Figure: The subsystem block diagram

Outline

BRADLEY University

V-REP Simulation

Background

- Commercial robot simulator
- Offers many robot models and supports multiple language interfaces
- Simulates the physics of the environment in addition to the robot

- V-REP is unable to simulate signal propagation so an alternate method was used
    1. Find ideal distance from model to beacon
    2. Convert ideal distance to dB · m
    3. Add noise using values in Table 1
    4. Convert dB · m back to m

Table: Simulation parameters used within V-REP.

Name	Value	Unit
σ_r	2	m
σ_β	18	$\circ$

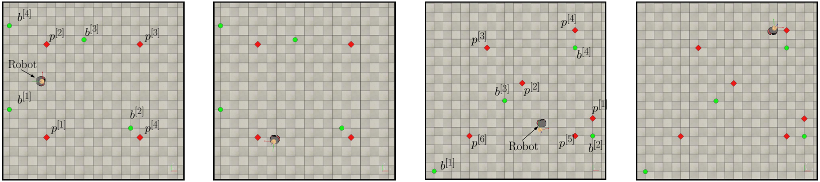
Outline

- Verify the EKF-SLAM algorithm before implementing
- Test the planned motion control strategy

Figure: Pioneer 3-DX model in available in V-REP

V-REP Simulation

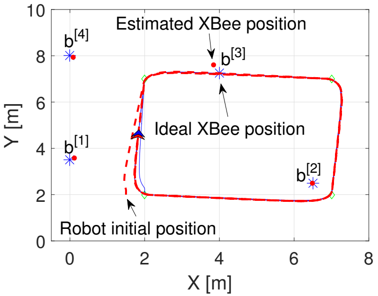
Simulation Cases



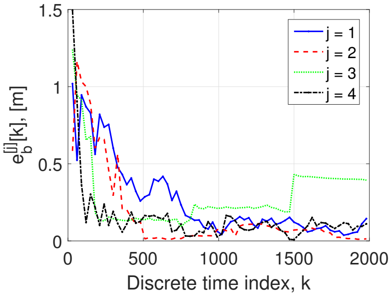
(a) (b) (c) (d)

Figure: Robot configurations in V-REP at (a) 25 s and (b) 300 s for case I. Robot configurations in V-REP at (c) 25 s and (d) 300 s for case II.

V-REP MATLAB



(a)



(b)

Figure: Partial results from Simulation Case I showing (a) the final trajectory, and (b) the beacon position estimation error.

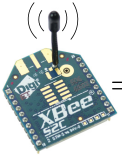
Outline

BRADLEY University

Customized Radio Transceiver

Range and Bearing Approximation

- Each XBee module is uniquely identified using the serial number *SL* field of the data packet
- Range is then calculated using the RSSI stored in *DB*
- RSSI is assumed to be the greatest when reflector is in the LoS of the XBee and this angle is determined to be the bearing



⇒

MY	SH	SL	DB	NI	\0
2	4	4	1	20	1

Figure: An example ZigBee data packet obtained using the *ND* command

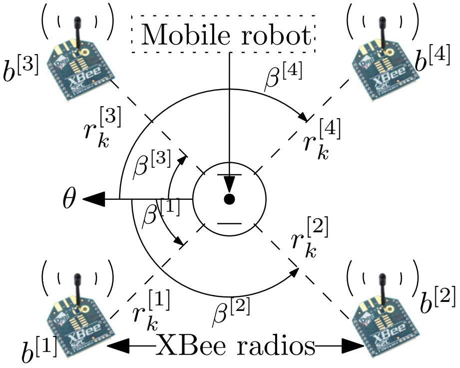


Figure: A diagram showing the range and bearing relationship.

Outline

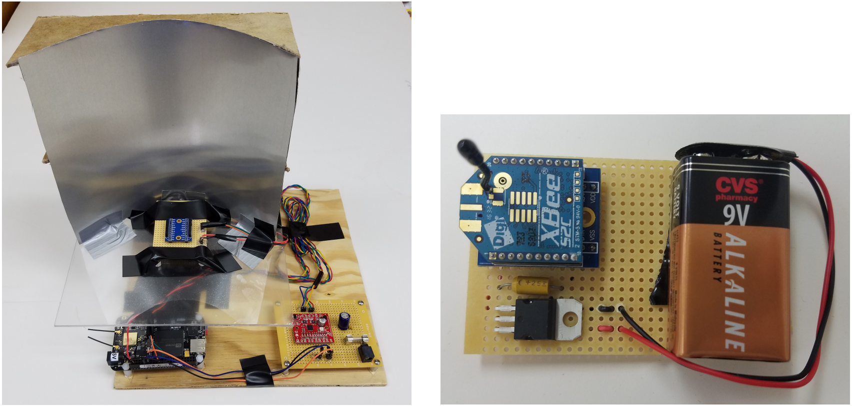
BRADLEY University

Customized Radio Transceiver

- Customized radio transceiver consisting of off the shelf components
  - BeagleBone Black Wireless
  - Stepper motor
  - Stepper motor controller
  - XBee S2C
- Total cost is around $140
- Beacons cost about $20
  - 9 V battery
  - XBee S2C
  - Voltage regulation circuitry

Customized Radio Transceiver

Hardware



(a) (b)

Figure: (a) The customized radio transceiver and (b) an assembled beacon

Customized Radio Transceiver

Hardware

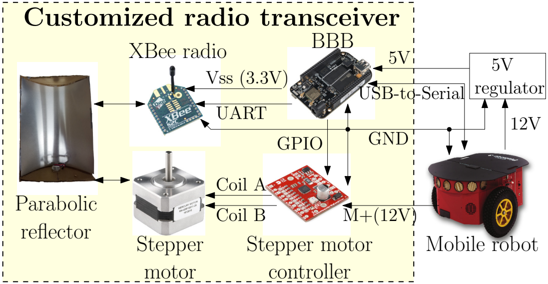


Figure: The connection diagram of the customized radio transceiver

- Code implemented in C++ and MATLAB
- The Robot Operating System (ROS) is used for communication between subsystems
  - More modularity
  - High–level abstraction from low–level operations

Outline

Table: Average of RSSI measurements obtained with customized radio transceiver

Ideal Range	0.5 m	1 m	2 m	3 m
Measured RSSI [dB · m]	−23	−33.103	−38.308	−41.974
Approximated Range[m]	0.3162	1.0119	1.8425	2.81
Error [m]	0.1838	−0.0119	0.1575	0.19

Table: Bearing approximation using customized radio transceiver

	$\mathbf{b}^{[1]}$	$\mathbf{b}^{[2]}$	$\mathbf{b}^{[3]}$	$\mathbf{b}^{[4]}$
Actual β	225°	120°	333°	45°
Measured β	207°	99°	330°	45°
Error	18°	21°	3°	0°

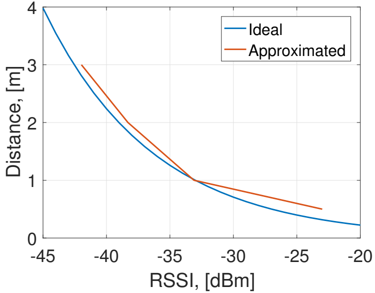


Figure: Distance vs RSSI comparing the ideal range from equation (4) and the approximate range from measurements

- Four beacons are placed at known positions to quantify the performance
- Inverse of equation (4) to determine range

$$r_k^{[j]} = 10^{\frac{|\mathrm{RSSI}| - |P_{\mathrm{ref}}|}{10\eta}}$$

- $P_{\mathrm{ref}} = -33$ dB · m and $\eta = 2$.

Customized Radio Transceiver

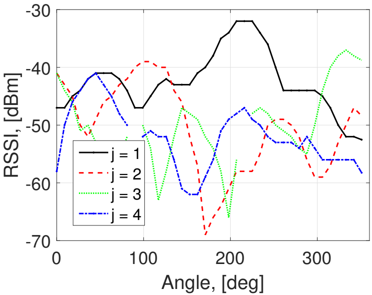


Figure: Data collected during the bearing estimation test

- For the bearing test, the robot was placed at $\mathbf{q} = [1\ \text{m}, 1\ \text{m}, 0\ \text{rad}]^T$, and beacons placed in the corners of the workspace at
  $\mathbf{b}^{[1]} = [0, 0]^T$ m,
  $\mathbf{b}^{[2]} = [0, 3]^T$ m,
  $\mathbf{b}^{[3]} = [3, 0]^T$ m, and
  $\mathbf{b}^{[4]} = [3, 3]^T$ m
- Reflector rotates at 9° increments for a full 360° and 40 RSSI measurements

Outline

BRADLEY University

Lab | MATLAB

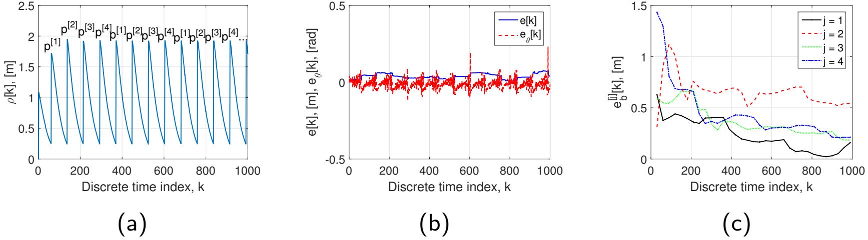


Figure: MATLAB plots showing (a) the fuzzy logic controller input, (b) the robot position estimation error, and (c) the beacon position estimation error

Lab MATLAB

Implementation

Experimental Case II

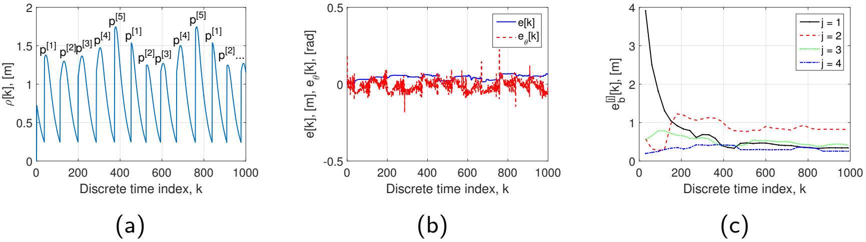


Figure: MATLAB plots showing (a) the fuzzy logic controller input, (b) the robot position estimation error, and (c) the beacon position estimation error

Implementation

Experimental Case III

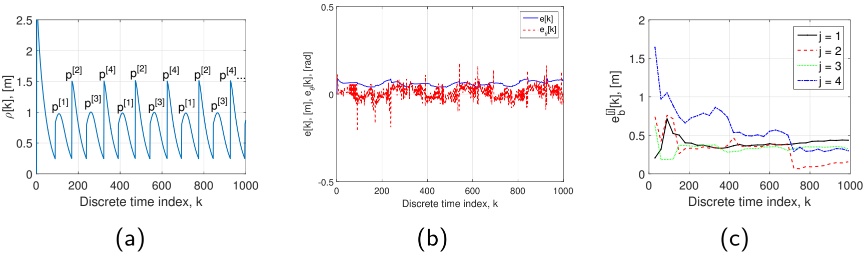


Figure: MATLAB plots showing (a) the fuzzy logic controller input, (b) the robot position estimation error, and (c) the beacon position estimation error

- Performance is adequate considering size of environment and number of beacons
- Most error is due to multipath in the environment but was minimal over shorter distances

Table: Performance comparison of experimental cases.

Case	RMSE [m]	RMSE_θ [rad]	$\mathrm{RMSE}_\mathrm{b}^{[1]}$ [m]	$\mathrm{RMSE}_\mathrm{b}^{[2]}$ [m]	$\mathrm{RMSE}_\mathrm{b}^{[3]}$ [m]	$\mathrm{RMSE}_\mathrm{b}^{[4]}$ [m]
I	0.04	0.03	0.28	0.67	0.39	0.53
II	0.05	0.04	1.03	0.88	0.53	0.33
III	0.06	0.04	0.41	0.36	0.34	0.66

Future Directions

- Improve the method of estimating the robot's pose
- Add some sort of filtering to improve the accuracy of range approximation
- Different methods of gathering measurements
- Adaptation for use in extreme environments

Conclusion

- The implemented EKF-SLAM algorithm performs modestly for short to medium range navigation and mapping in noisy environments
- Completed work under the budget of $500
- Robot's final pose is estimated within 30 cm of its true position and some XBee radios final estimated positions are within 20 cm
- Due to recursive nature of algorithm, longer experimental times will reduce estimation error

Thank you for your attention