



PORTFOLIO

(na wniosek firmy F16)

Możliwości pozycjonowania wewnątrz budynków z wykorzystaniem radiolatarni

Autor: Sebastian Ernst, Piotr Matyasik

Centrum Inteligentnych Systemów Informatycznych Akademia Górniczo-Hutnicza im. Stanisława Staszica al. Mickiewicza 30, 30-059 Kraków
budynek C-2 pokój 426 tel: 12 617 44 53 www.isi.agh.edu.pl isi@agh.edu.pl



KAPITAŁ LUDZKI
INICJATYWA INTELIGENTNYCH SYSTEMÓW



Możliwości pozycjonowania wewnątrz budynków z wykorzystaniem radiolatarni

dr inż. Piotr Matyasik
dr inż. Sebastian Ernst

Wersja: 2014-04-30 (796f81a)

Spis treści

1	Wprowadzenie	2
2	Opis standardu iBeacon	2
2.1	Wsparcie w platformach mobilnych	2
2.1.1	Apple iOS	2
2.1.2	Android	3
2.1.3	Windows Phone	3
2.2	Możliwości programowego generowania sygnałów	4
2.3	Dostępne na rynku urządzenia-radiolatarnie	4
3	Charakterystyka przeprowadzonych badań	4
3.1	Zaimplementowane aplikacje testowe	4
3.2	Scenariusze testów	4
4	Wyniki i analiza	6
5	Podsumowanie	12

1 Wprowadzenie

Przez ostatnie kilka miesięcy obserwuje się wzrost zainteresowania technologią tzw. radiolatarni – nadajników radiowych, najczęściej opartych o standard Bluetooth, emitujących sygnały dające się odebrać przez urządzenia mobilne, takie jak smartfony oraz tablety.

Technologia ta pozwala na dodanie do aplikacji świadomości lokalizacji (ang. *location awareness*) nie tylko na zewnątrz (gdzie dostępny jest system GPS oraz technologie wspomagające), ale i wewnątrz budynków. Podstawowym zadaniem radiolatarni jest wychwycenie znalezienia się urządzenia w pewnym otoczeniu nadajnika, choć teoretycznie możliwe jest też wykorzystanie tej technologii do określania lokalizacji urządzenia na planie pomieszczeń (a także np. wykrywanie piętra).

Niniejszy raport przedstawia wstępne wyniki badań dotyczących możliwości lokalizowania urządzeń mobilnych przy pomocy radiolatarni.

2 Opis standardu iBeacon

Standard iBeacon został wprowadzony przez firmę Apple w wersji 7 mobilnego systemu operacyjnego iOS. Standard oparty jest o technologię Bluetooth Low Energy (BLE). Nadajniki charakteryzowane są przez kilka parametrów:

- *proximity UUID* – identyfikator, który jest identyczny dla wszystkich radiolatarni zainstalowanych w jednej lokalizacji¹,
- *major ID* – liczba całkowita, grupująca radiolatarnie na pierwszym poziomie hierachii (np. piętro, dział),
- *minor ID* – liczba całkowita, identyfikująca poszczególne radiolatarnie w obrębie danej grupy.

Aby umożliwić oszacowanie odległości, radiolatarnie podają również dane kalibracyjne – poziom sygnału (w dB) w odległości 1 metra od nadajnika.

W przestrzeni wyposażonej w radiolatarnie możliwe jest prowadzenie dwóch operacji:

1. Monitorowanie regionów – notyfikacja aplikacji, gdy urządzenie znajdzie się w obrębie regionu, przy czym region może być identyfikowany przez:
 - sam *proximity UUID*,
 - *proximity UUID* oraz *major ID*,
 - wszystkie trzy parametry (*proximity UUID*, *major ID* oraz *minor ID*).
2. Namierzanie radiolatarni (ang. *beacon ranging*) – skanowanie pasma w celu lokalizacji „widocznych” radiolatarni, przy czym konieczne jest podanie co najmniej parametru *proximity UUID*.

2.1 Wsparcie w platformach mobilnych

2.1.1 Apple iOS

Wsparcie obsługi radiolatarni zgodnej ze standardem iBeacon zostało wprowadzone w wersji 7.0 systemu iOS. Centralną rolę gra tu framework *Core Location*, który udostępnia metody opakowujące z kolei funkcjonalność frameworku *Core Bluetooth*.

Dostępne są metody realizujące obie opisane w sekcji 2 funkcjonalności. Monitorowanie regionów realizowane jest w trybie asynchronicznym i inicjowane jest przy pomocy metody `startMonitoringForRegion:` obiektu `CLLocationManager`. W przypadku wejścia i wyjścia z regionu wywoływane są odpowiednio metody `locationManager:didEnterRegion:` oraz `locationManager:didExitRegion:`.

Skanowanie pasma inicjowane jest przy pomocy metody `startRangingBeaconsInRegion:`. Po uzyskaniu nowych danych wywoływana jest metoda `callback locationManager:didRangeBeacons:inRegion:`, która otrzymuje jako jeden z parametrów tablicę składającą się z obiektów klasy `CLBeacon`. Każdy z nich posiada następujące informacje:

- dane identyfikacyjne – `proximityUUID`, `major`, `minor`,

¹Przez lokalizację uznaje się obszar (np. teren sklepu), zarządzany przez jedną instytucję, na którym prowadzone jest pozycjonowanie i/lub wykrywanie bliskości przez daną aplikację mobilną.

Stan	Maksymalny czas aktualizacji
Aktywne skanowanie (<i>ranging</i>)	1 sekunda
Skanowanie wyłączone	do 15 minut

(a) Czasy aktualizacji dla aplikacji pracującej w pierwszym planie

Stan	Maksymalny czas aktualizacji
Telefon obudzony, <code>notifyEntryStateOnDisplay=YES</code>	1 sekunda
Telefon obudzony, <code>notifyEntryStateOnDisplay=NO</code>	nigdy
<code>UIBackgroundModes=location ON</code>	do 15 minut
<code>UIBackgroundModes=location OFF</code>	do 15 minut

(b) Czasy aktualizacji dla aplikacji pracującej w tle

Tablica 1: Czasy wykrywania radiolatarni w platformie iOS [3]

- `rs` – siła sygnału odebranego, mierzona w dB,
- `proximity` – klasa bliskości; typ wyliczeniowy mogący przybierać następujące wartości [1]:
 - `CLProximityUnknown` – jeżeli odległość nie może być oszacowana,
 - `CLProximityImmediate` – radiolatarnia w bezpośredniej bliskości użytkownika,
 - `CLProximityNear` – radiolatarnia względnie blisko użytkownika,
 - `CLProximityFar` – radiolatarnia zlokalizowana daleko,
- `accuracy` – szacowana odległość w metrach od radiolatarni².

Częstotliwość odświeżania odpowiednio przy aplikacji pracującej na pierwszym planie i w tle przedstawiono (dla różnych wartości parametrów konfiguracyjnych) w Tab. 1. Programista nie ma możliwości ingerowania w te parametry.

2.1.2 Android

API dla Androida jest kopią API ogłoszonego dla iOS. W teście została wykorzystana implementacja wykonana przez <http://developer.radiusnetworks.com>.

API pozwala na:

- powiadamianie, jakie nadajniki są widoczne,
- ustawienie częstotliwości powiadamiania o widocznych nadajnikach przy wybudzonym urządzeniu,
- ustawienie częstotliwości powiadamiania o widocznych nadajnikach przy uśpionym urządzeniu,
- powiadamianie o wejściu w wybraną strefę (*Region*),
- powiadamianie o osiągnięciu wybranej pozycji w strefie (`PROXIMITY_IMMEDIATE`, `PROXIMITY_NEAR`, `PROXIMITY_FAR`, `PROXIMITY_UNKNOWN`),
- uruchamianie aplikacji po znalezieniu się w danej odległości w wybranej strefie.

Usługa systemowa uśrednia sygnał RSSI, nie wykonuje żadnej dodatkowej obróbki.

2.1.3 Windows Phone

Przewidywane jest wsparcie w Windows Phone 8.1.

²Według dokumentacji, wartość ta służy wyłącznie do rozróżniania pomiędzy radiolatarniami z identycznym parametrem `proximity`. Nie należy jej używać do oszacowania dokładnej lokalizacji radiolatarni.

2.2 Możliwości programowego generowania sygnałów

Rolę radiolatarni może pełnić dowolne urządzenie wyposażone w moduł Bluetooth 4.0 LE. Przykładowo, możliwe jest bardzo łatwe „przekształcenie” w radiolatarnię urządzenia pracującego pod kontrolą systemu iOS lub Mac OS X, korzystając z metod udostępnianych przez framework Core Bluetooth.

2.3 Dostępne na rynku urządzenia-radiolatarnie

Szybki wzrost zainteresowania technologią spowodował pojawienie się na rynku licznych urządzeń pełniących funkcję radiolatarni zgodnych ze standardem iBeacon. Są to m.in.:

- <http://estimote.com/> - szczegółowy opis znajduje się pod adresem <http://makezine.com/2014/01/03/reverse-engineering-the-estimote/>
- <http://buyibeacons.com/>
- dowolny adapter Bluetooth 4.0 wspierający BLE z oprogramowaniem

3 Charakterystyka przeprowadzonych badań

W badaniach wykorzystano radiolatarnie bazujące na adapterach Bluetooth 4.0 BLE z oprogramowaniem *bluez-ibeacon* <https://github.com/carsonmcdonald/bluez-ibeacon> oraz iPad.

Przyjęto następujący model zanikania sygnału:

$$L(dB) = P_0 + 10 n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma$$

gdzie: n - współczynnik straty sygnału, d - odległość pomiędzy nadajnikiem i odbiornikiem, X_σ - gausowska zmienna losowa o odchyleniu standardowym σ i P_0 moc odbierana w odległości d_0 .

Dla RSSI, przy założeniu $d_0 = 1m$ i pominięciu X_σ ogranicza się to do:

$$RSSI = -10 n \log_{10}(d) + P_{1m}$$

gdzie P_{1m} jest mocą odbieraną w odległości 1m.

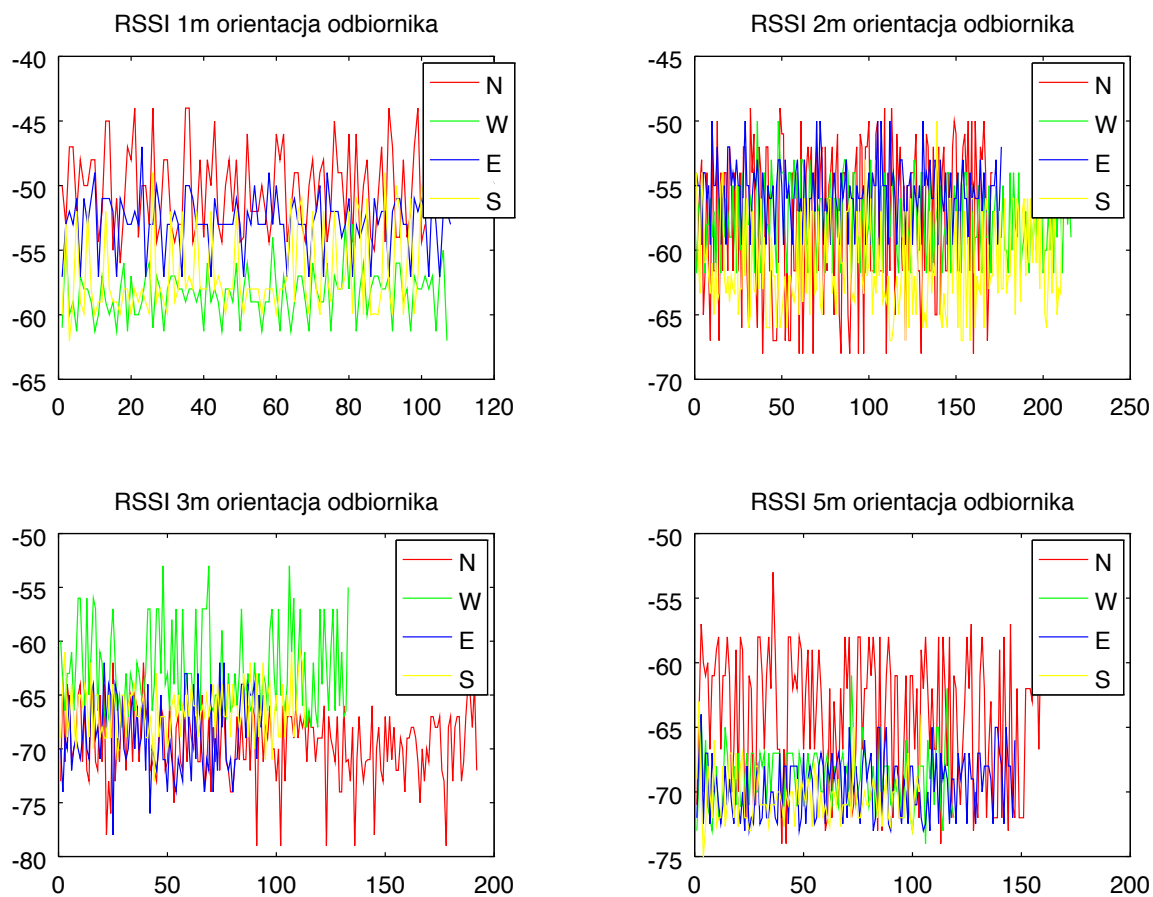
3.1 Zaimplementowane aplikacje testowe

Dla systemu Android została opracowana aplikacja monitorująca radiolatarnie na bazie serwisu dostarczonego przez <http://developer.radiusnetworks.com>. W trakcie działania wyświetla ona i zapisuje do pliku widoczne radiolatarnie.

3.2 Scenariusze testów

Poniżej znajdują się opisy przeprowadzonych testów:

1. estymacja parametrów dla modelu zanikania sygnału dla różnych nadajników,
2. badanie zależności wzajemnej orientacji nadajnika i odbiornika na parametry modelu,
3. badanie wpływu sposobu trzymania odbiornika na poziom RSSI,
4. badanie wpływu innych osób na poziom RSSI,
5. określenie odległości od nadajników na podstawie poprzednio estymowanych parametrów modelu.



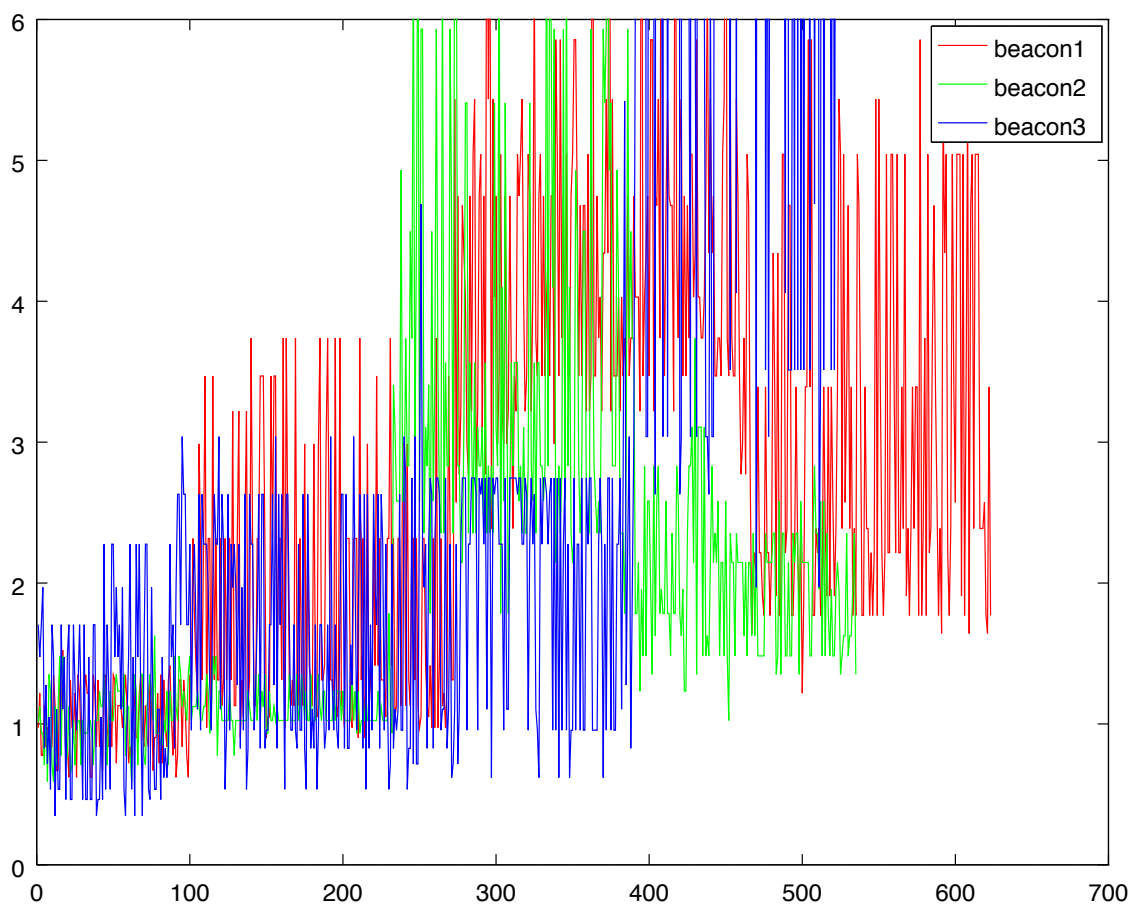
Rysunek 1: RSSI w zależności od orientacji i odległości dla wybranego nadajnika

Nadajnik	oszacowanie n
Beacon1	3.0789
Beacon2	2.4915
Beacon3	1.5921

Tablica 2: Oszacowanie parametru n modelu dla kierunku N

Nadajnik	N	W	E	S
Beacon1	-50.364	-58.370	-52.835	-56.800
Beacon2	-55.737	-60.589	-56.356	-56.087
Beacon3	-48.314	-47.611	-55.805	-53.269

Tablica 3: Oszacowanie parametru A modelu w zależności od kierunku i nadajnika



Rysunek 2: Oszacowana odległość bez filtracji

4 Wyniki i analiza

Wykresy 1 przedstawiają siłę sygnału doczytywaną przez urządzenie w zależności od orientacji względem nadajnika.

Tabela 3 przedstawia uśrednioną wartość sygnału w odległości 1m od nadajników w zależności od orientacji urządzenia. Przed uśrednianiem z próbki pomiarowej usunięte zostały te wartości, które odbiegały od średniej o więcej niż 20%. Miało to na celu pozbycie się pomiarów, które najprawdopodobniej były wynikiem odbić sygnału.

Tabela 2 przedstawia oszacowaną wartość parametru n modelu dla wszystkich czujników w położeniu N. Oszacowanie przeprowadzone zostało z wykorzystaniem metody najmniejszych kwadratów. Należy podkreślić, że użyte nadajniki były tego samego modelu i zachowano identyczne warunki pomiarowe dla każdego z nich.

Wykres 2 przedstawia oszacowaną odległość bez filtracji.

Wykres 3 przedstawia oszacowaną odległość filtrowaną metodą średniej ruchomej. Pomiarów były wykonywane w odległości 1,2,3 i 5 metrów.

Wykres 4 przedstawia oszacowaną odległość filtrowaną metodą średniej ruchomej, uśrednioną po wszystkich nadajnikach.

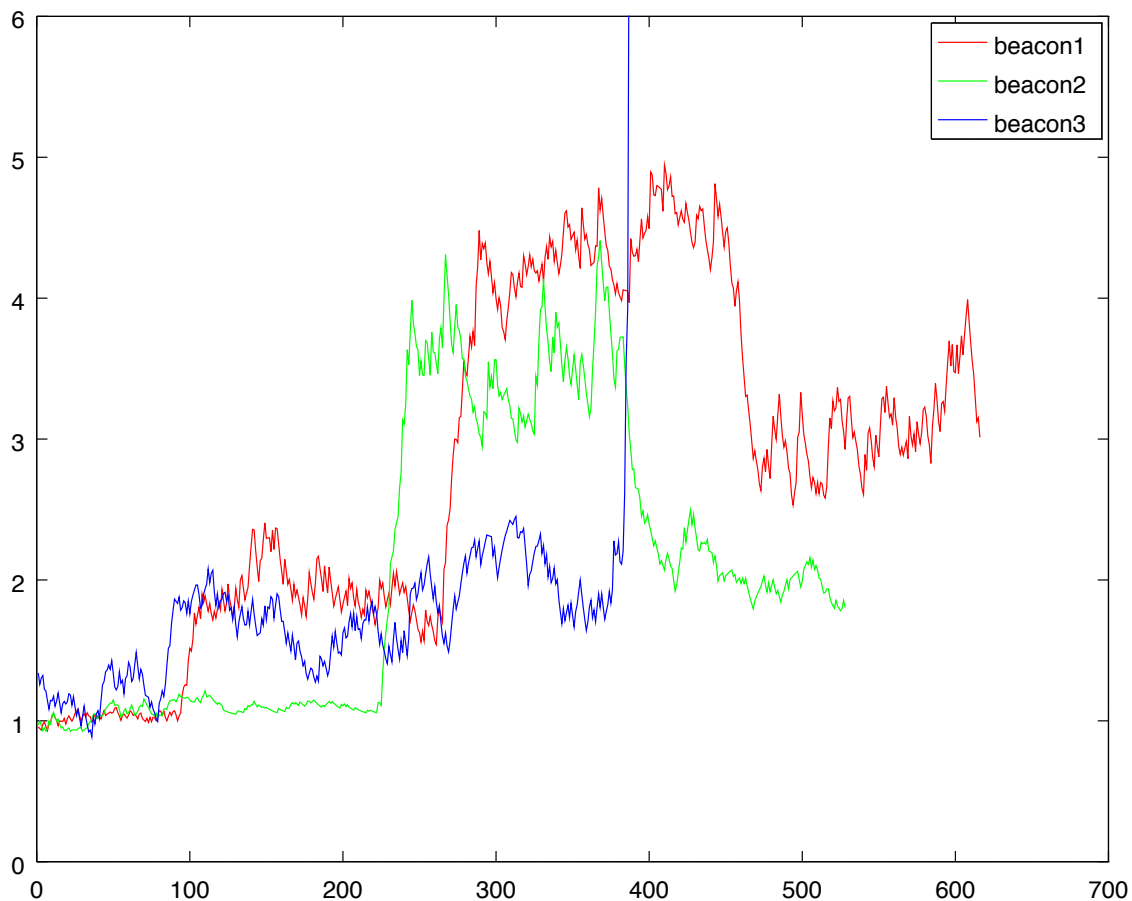
Wykres 5 przedstawia oszacowaną odległość filtrowaną metodą średniej ruchomej, uśrednioną po wszystkich nadajnikach. Dodatkowo wartości wygładzono metodą Tichonowa.

Wykres 6 przedstawia zmianę średniej wartości RSSI wynikającą tylko ze zmiany sposobu trzymania odbiornika w rękach. Wzajemna orientacja odbiornika i nadajnika oraz odległość pozostawały bez zmian.

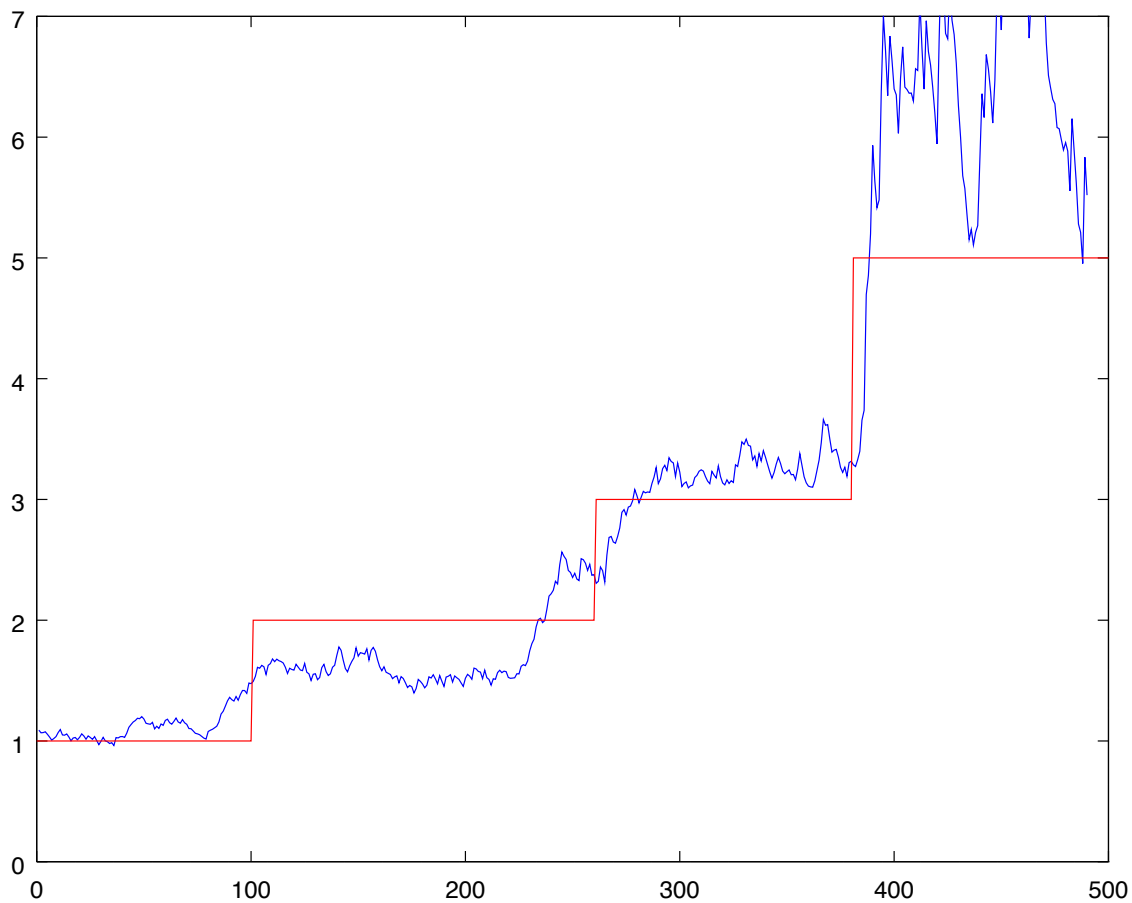
Wykres 6 przedstawia zmianę średniej wartości RSSI wynikającą z pojawiania się i znikania osoby pomiędzy nadajnikiem i odbiornikiem.

Wykres 8 wynik pomiaru RSSI dla testu polegającego na przemieszczeniu urządzenia z odległości 5m do 1m i na powrót do 5m.

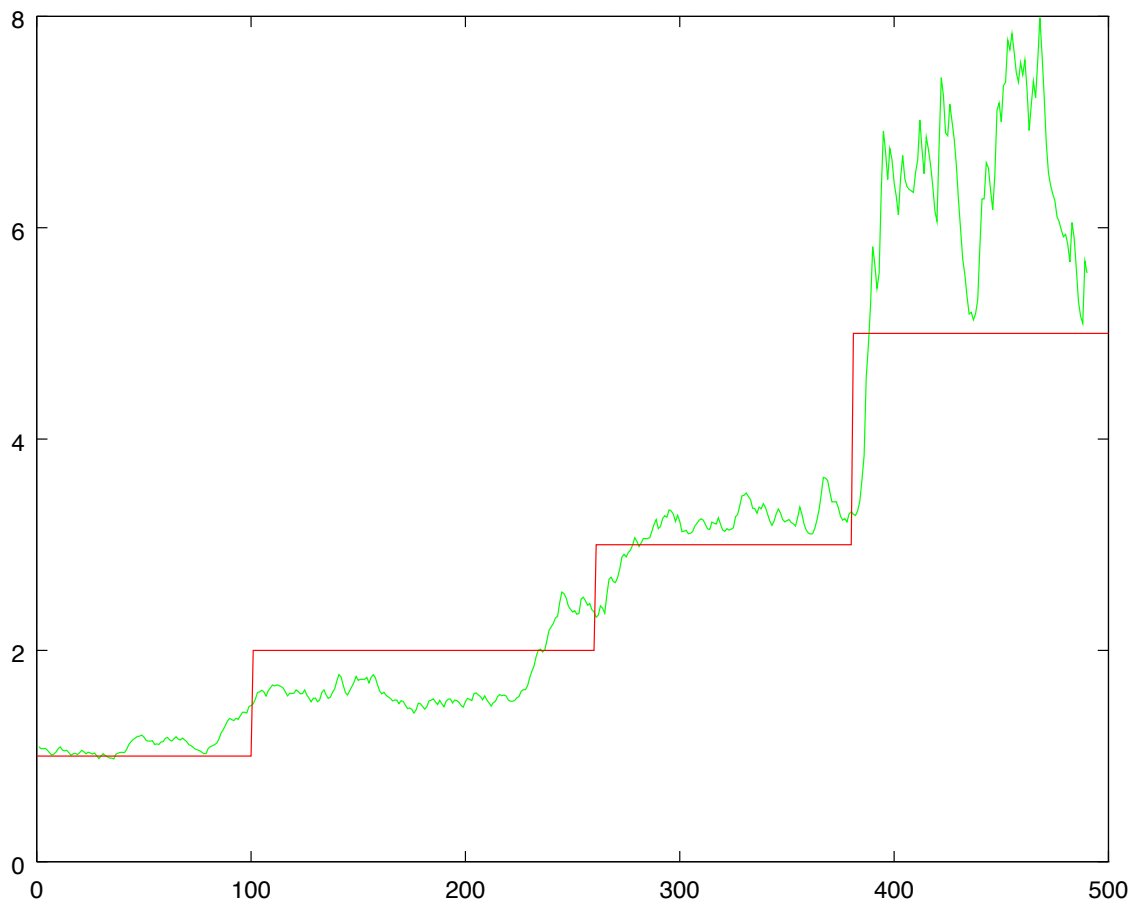
Wykresy 2 do 12 przedstawiają estymacje odległości filtrowane analogicznie jak w poprzednich przypadkach.



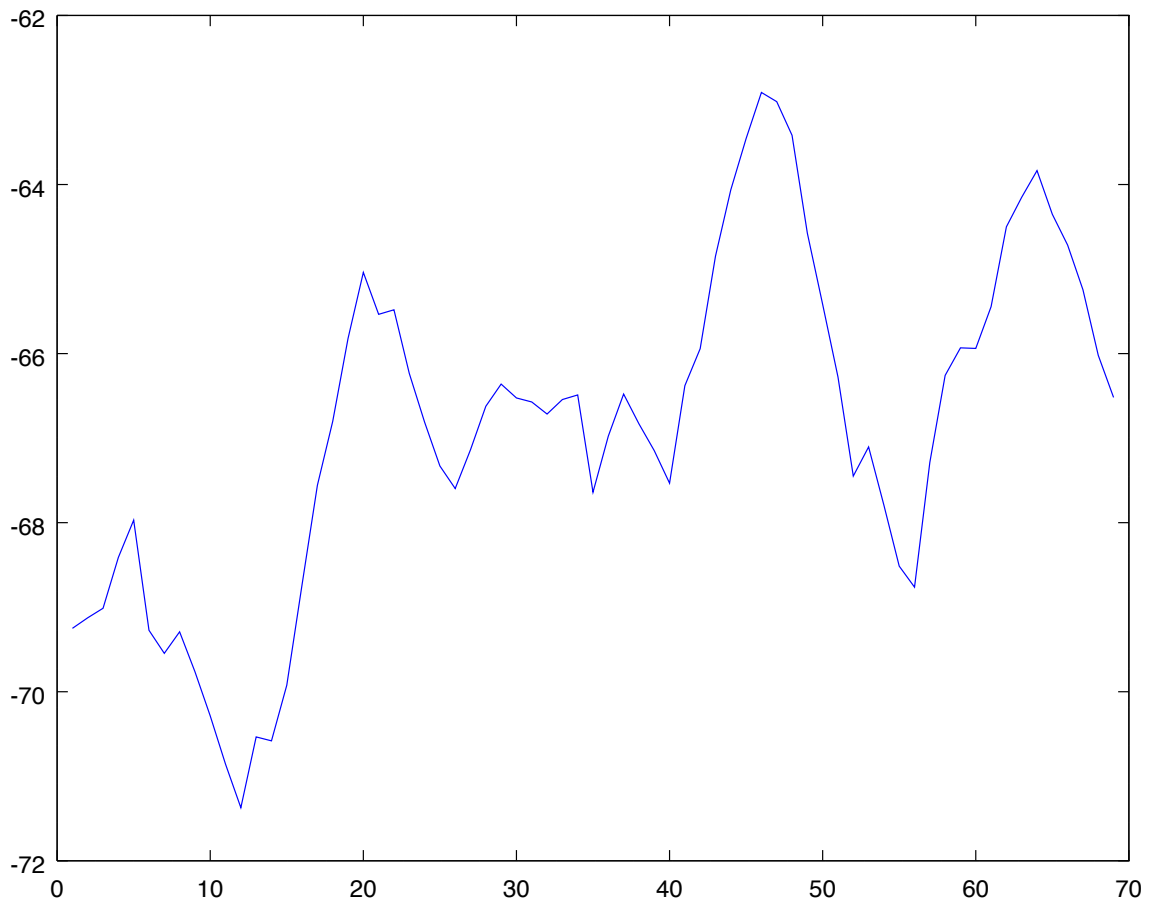
Rysunek 3: Oszacowana odległość (średnia ruchoma)



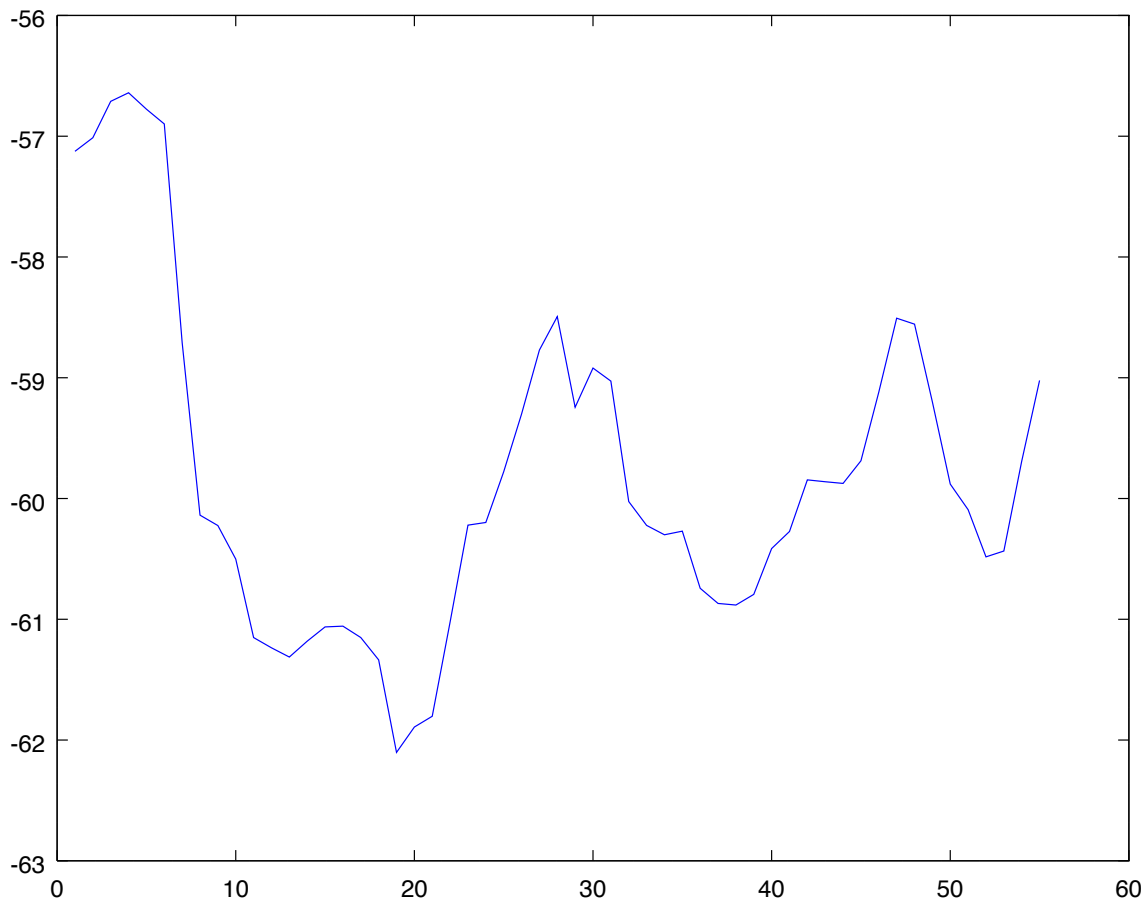
Rysunek 4: Oszacowana odlegość (średnia wszystkich nadajników)



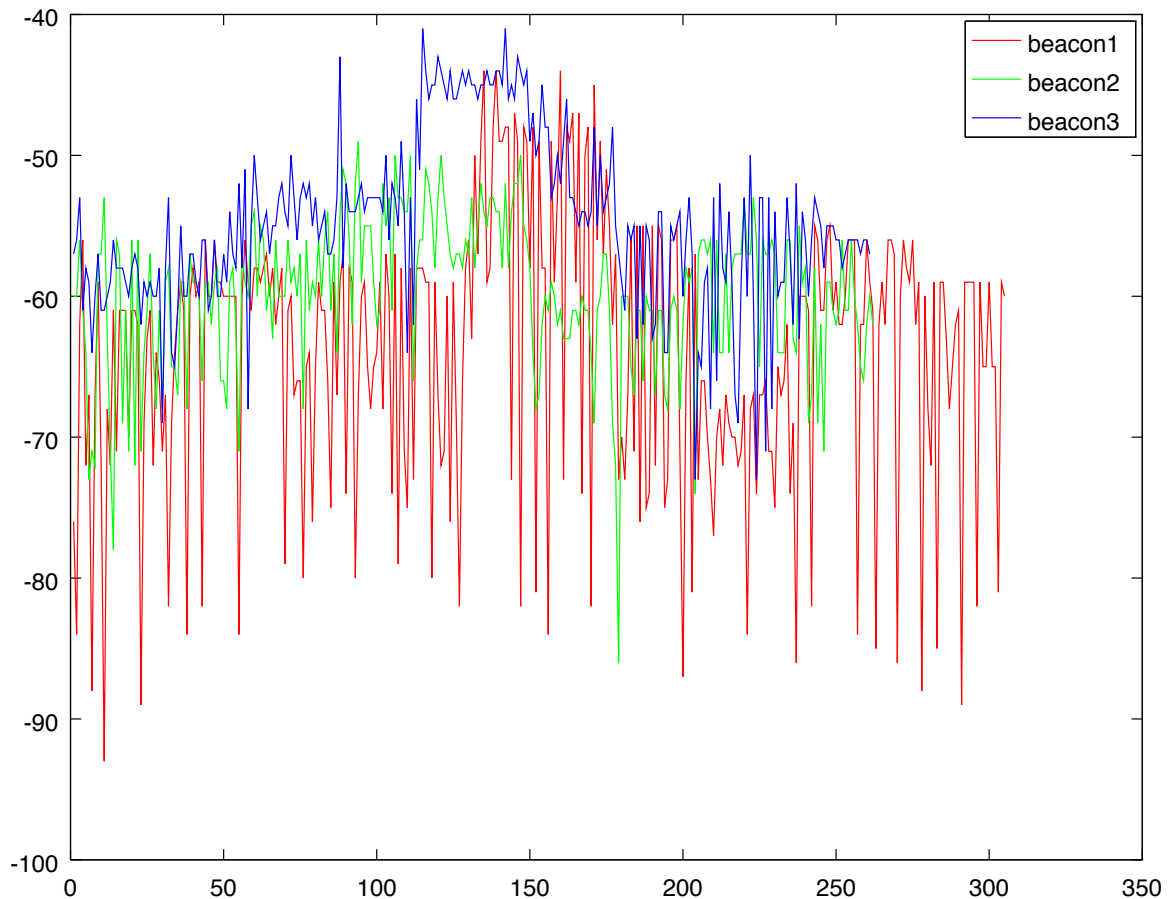
Rysunek 5: Oszacowana odległość (średnia wszystkich nadajników + Tichonow)



Rysunek 6: Średni RSSI przy różnym sposobie trzymania odbiornika



Rysunek 7: Średni RSSI - osoba pojawiająca się pomiędzy nadajnikiem i odbiornikiem



Rysunek 8: RSSI - test w ruchu

5 Podsumowanie

Pozycjonowanie urządzeń wewnątrz budynków pomimo wykorzystania różnych, podstawowych rodzajów filtrowania daje umiarkowanie zadowalające rezultaty. Uśredniona wartość błędu dla testu w ruchu oscyluje w granicy 1m. Pomiary dokonywane były z zachowaniem stałej orientacji urządzeń względem siebie.

Zastosowanie kilku urządzeń podnosi użyteczność informacji, jaką można uzyskać. Wymaga to jednak dokładnej kalibracji każdego z nich. Pomimo użycia nadajników tego samego typu ich parametry znacznie się różniły.

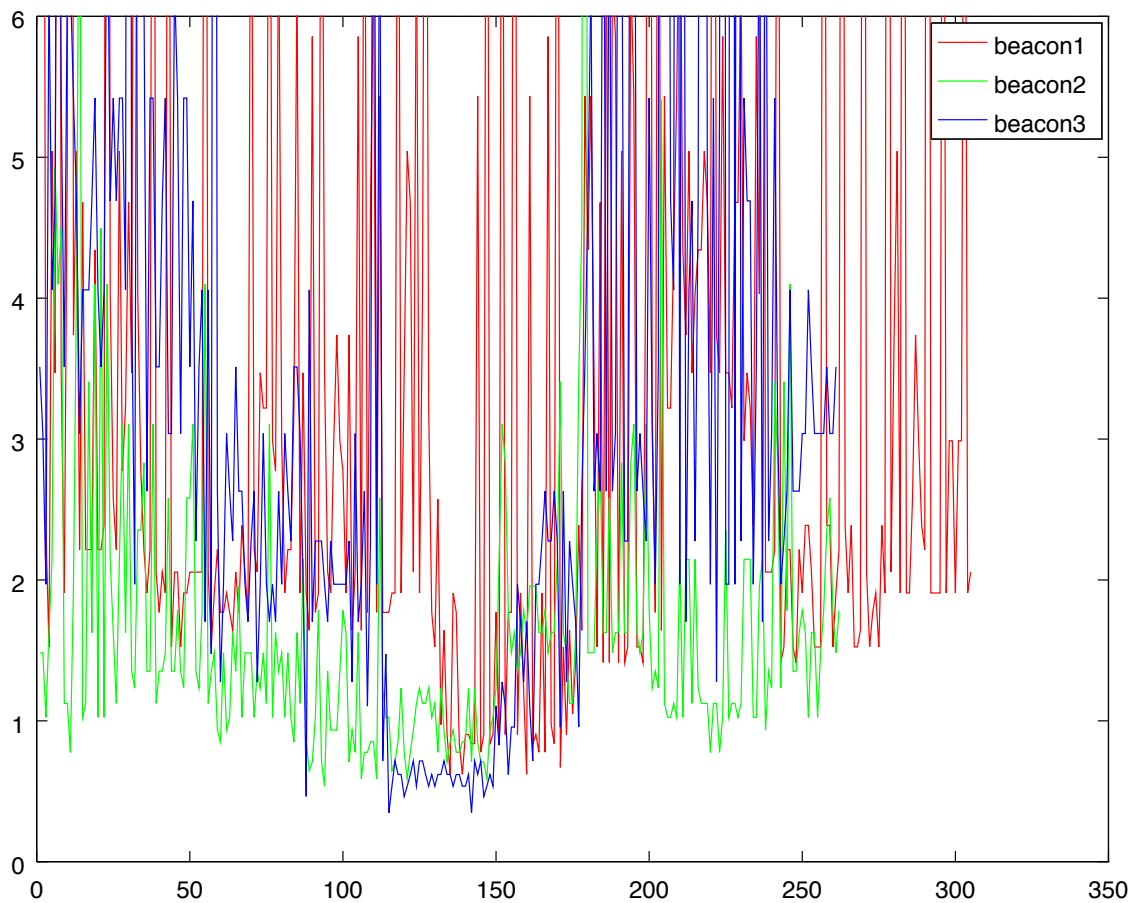
Zastosowane metody filtracji były różnymi wariantami średniej i dawały pozytywne rezultaty. Ich użycie jest konieczne ze względu na duży poziom szumu w sygnale, ale ogranicza to użycie tej metody tylko do szacowania odległości obiektów poruszających się stosunkowo wolno.

Odległość podawana przez API urządzenia jest całkowicie bezużyteczna. Zależy ona od emitowanego parametru A (RSSI w odległości 1m), który może być różny w zależności od modelu urządzenia odbierającego, użytych akcesoriów, sposobu noszenia, wzajemnej orientacji nadajnika i odbiornika.

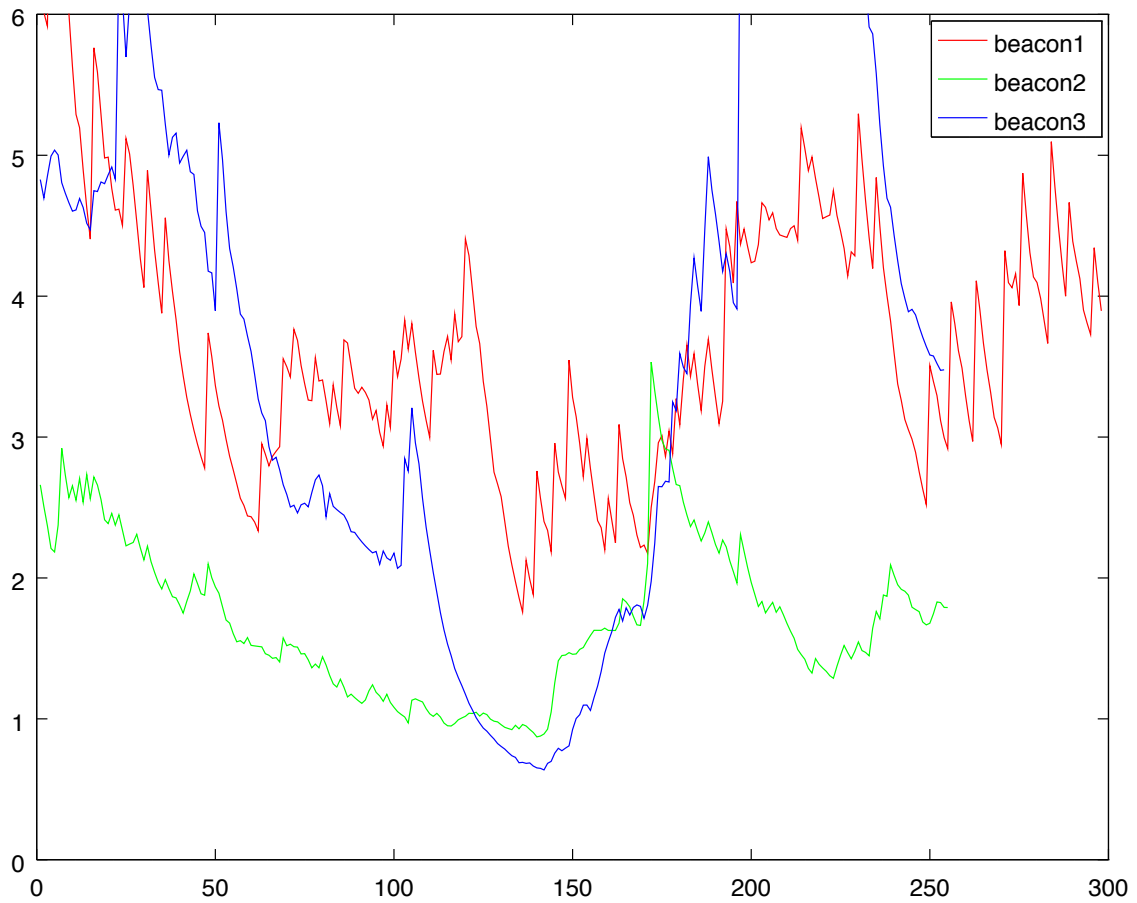
Stopień zbliżenia (proximity type) w danej strefie jest określany na podstawie RSSI, stąd występują częste przeskoki na granicy stref wynikające ze zmiany położenia, znalezienia się osób lub obiektów na linii nadajnik-odbiornik.

Ewentualne dalsze kierunki badań mogące podnieść użyteczność technologii:

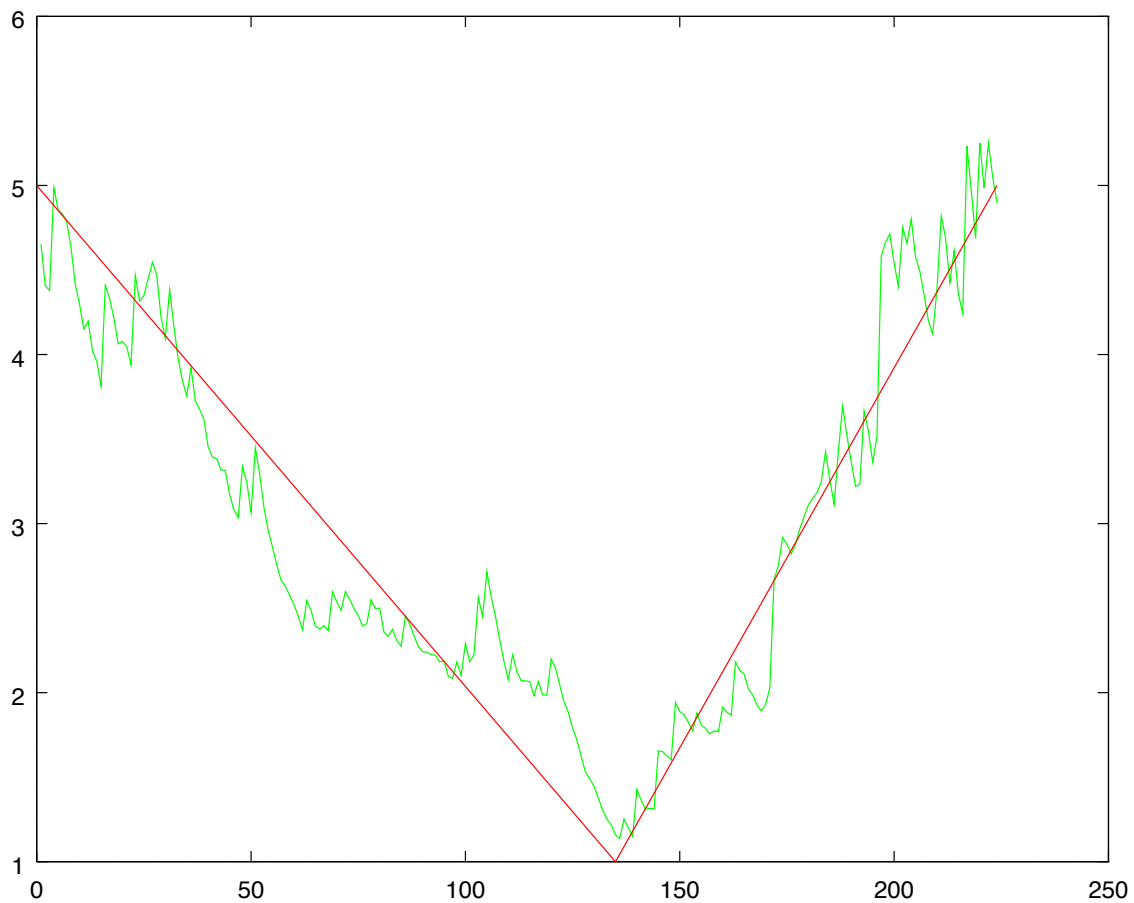
- opracowanie modelu zanikania sygnału uwzględniającego wzajemne położenia nadajnika i odbiornika,
- uwzględnienie w filtrze dodatkowych informacji (akcelerometr, żyroskop, magnetometr),
- uwzględnienie wiedzy o wzajemnym położeniu kilku nadajników [2],



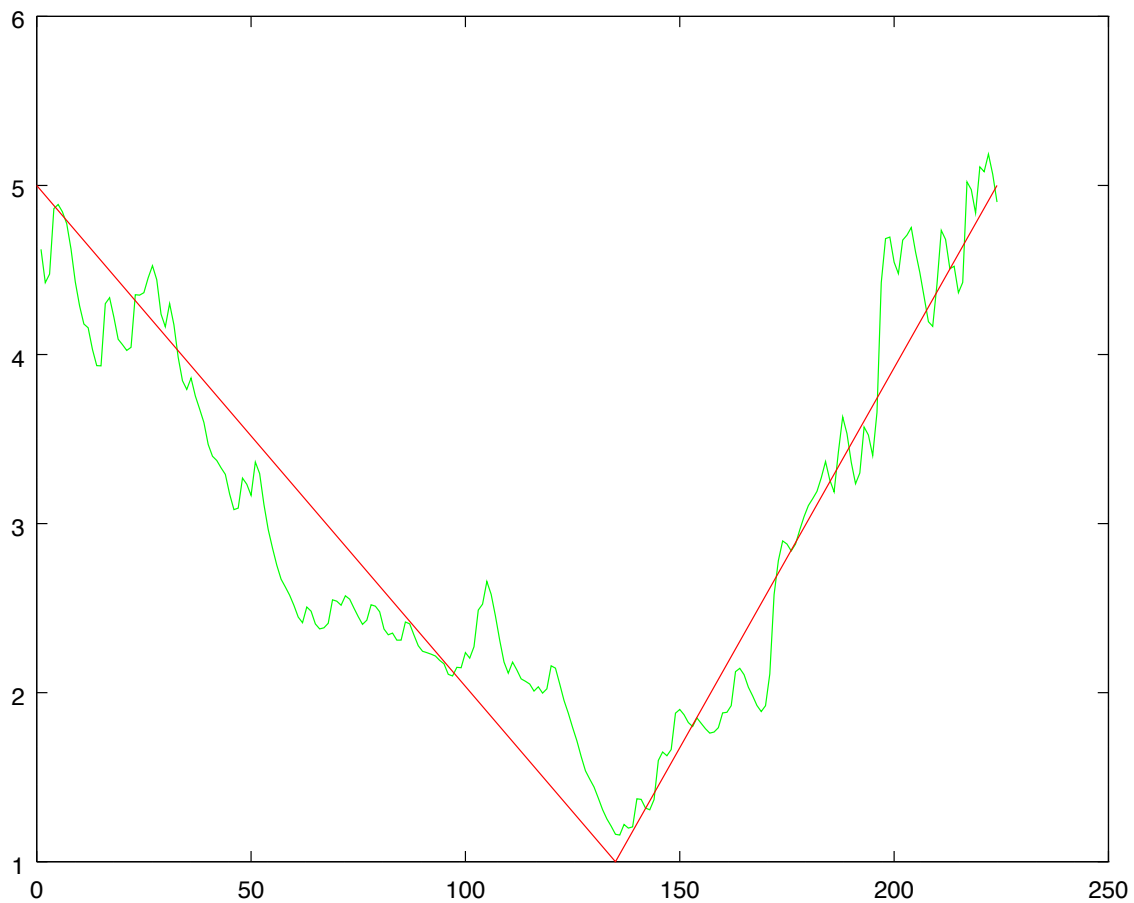
Rysunek 9: Odległość - test w ruchu



Rysunek 10: Odległość - test w ruchu - średnia ruchoma



Rysunek 11: Odległość - test w ruchu - średnia wszystkich nadajników



Rysunek 12: Odległość - test w ruchu - średnia wszystkich nadajników wygładzona m. Tichonova

- uwzględnienie wiedzy o otoczeniu (rozmiar pomieszczenia, przeszkody, itp.),
- opracowanie zasad rozmieszczania nadajników tak, aby zminimalizować ewentualne zakłócenia.

Literatura

- [1] Apple, Inc. CLBeacon class reference. https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CLBeacon_class/Reference/Reference.html#/apple_ref/occ/cl/CLBeacon, 2014.
- [2] Jagoba Arias, Aitzol Zuloaga, Jesús Lázaro, Jon Andreu, and Armando Astarloa. Malguki: an {RSSI} based ad hoc location algorithm. *Microprocessors and Microsystems*, 28(8):403 – 409, 2004. Resource Management in Wireless and Adhoc mobile networks.
- [3] David G. Young. iBeacon monitoring in the background and foreground. <http://developer.radiusnetworks.com/2013/11/13/ibeacon-monitoring-in-the-background-and-foreground.html>, November 2013.