**ניסיונות למדידת מרחק בעזרת WIFI**

לפי [זה](https://www.mdpi.com/2076-3417/10/6/2003/pdf) (עמוד 5, נוסחה (3)) ומקורות נוספים שמצאנו, ניתן לקבל RSSI ובאמצעותו למדוד את המרחק (d) בהתאם לנוסחה הבאה:

כאשר הוא ה-RSSI המתקבל בנקודת "ייחוס" הנמצאת במרחק ממקור ה-WIFI.

עבור *, נסמן*

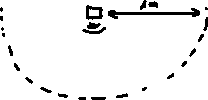
*ונקבל*

ניסינו לבצע מספר ניסוייים על מנת למצוא את הפרמטרים A, ו-n ולחשב את המרחק.

**ניסוי ראשון:**

"מקור" ה-WIFI הוחזק במקום אחד, ובקר נוסף שמודד RSSI, "הסתובב" סביבו בכחצי מעגל ברדיוס מסוים, על מנת לבדוק מה ה-RSSI המתקבל במרחק זה מכמה כיוונים. המדידות מתבצעות במרווחי זמן קבועים של 20 מילישניה.

שרטוט להמחשה:



חזרנו על הניסוי במרחקים: 0.5 מטר, 1 מטר, 2 מטרים, 3 מטרים, 4 מטרים ו-5 מטרים.

התוצאות שהתקבלו (התעלמנו מחלק מהדגימות האחרונות בחלק מהמרחקים, על מנת לשמור על אחידות במספר הדגימות):

התקבלו תוצאות לא מאוד יציבות ועם הרבה קפיצות, למרות שהמרחק היה פחות או יותר קבוע עבור כל עקומה. ניסינו לחפש דרכים להפוך את המדידה ליותר "יציבה".

כאן: <https://journals.sagepub.com/doi/pdf/10.1155/2015/195297> מופיעות מספר אפשרויות לשיפור התוצאות. נכון לעכשיו, ולפי [הצעות נוספות](https://www.wouterbulten.nl/blog/tech/kalman-filters-explained-removing-noise-from-rssi-signals/) שהופיעו באינטרנט, ניסינו להשתמש במימוש של Kalman filter (קוד שנלקח מכאן: <https://github.com/bachagas/Kalman/blob/master/README.md>).

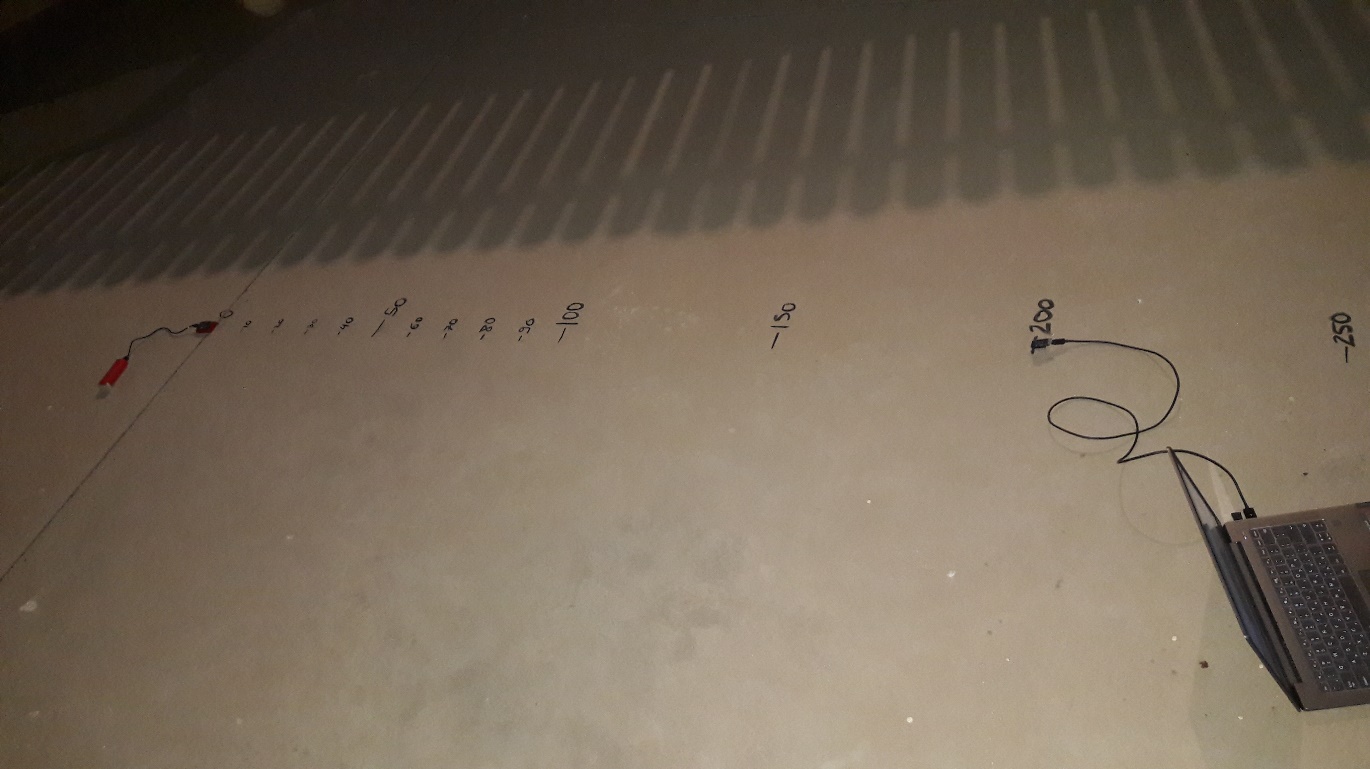
כך נראית ה­­השפעה שלו, למשל עבור מרחק של מטר אחד:

ועבור כל המרחקים שנבדקו:

העקומות יותר "חלקות" בדרך הזו.

ה-RSSI המתקבל אמור להיות מונוטוני כפונקציה של המרחק, וניתן לראות שזה לא קרה הפעם (העקומה עבור מרחק חצי מטר נמצאת בין 1 ל-2, למשל).

כלומר עדיין, ניתן לראות שיש הרבה רעש והמדידה לא מדויקת כל כך, ולכן ביצענו ניסוי נוסף:

"מקור" ה-WIFI מונח במקומו, ואנו מודדים את ה-RSSI ממרחקים שונים (באותו הכיוון), כך שבכל שלב הבקר המודד נמצא בנקודה קבועה ולא זז. תמונה להמחשה: 

חישבנו את הפרמטר n בתור ה-n הממוצע שהתקבל מהמדידות כאשר המרחק ידוע, בהתאם לנוסחה, ולאחר מכן השתמשנו בערך שהתקבל כדי לחשב את המרחקים.

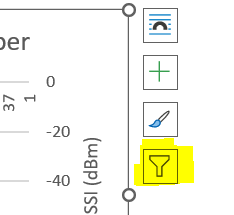
התוצאות שהתקבלו:

התוצאות שהתקבלו יחד עם הfilter-, והתוצאות שהתקבלו מה-RSSI הרגיל מאוד קרובות במקרה זה, שבו האות נקלט בלי תזוזה של הבקר (המדידות היו די יציבות עבור מרחק קבוע ללא תזוזה). השגיאה היחסית ( ) הממוצעת שקיבלנו עבור RSSI רגיל היא 32.3%, לעומת 32.2% עבור הfiltered RSSI.

כמו כן, עבור מרחקים גדולים יותר, החל מ-2 מטרים, השגיאה שהתקבלה היא 43.1% ו-42.9% בהתאמה. כלומר יש שיפור קטן (מאוד), אך עדיין שגיאה גדולה.

נרצה לבחון את ההשפעה של ה-filter שניסינו, וגם אפשרויות נוספות ל"סינון", כאשר הבקר לא לגמרי במנוחה, למשל מוחזק ביד במקום, כפי שאמור להיות במשחק עצמו.

**ניסוי נוסף:** התקרבות בקצב פחות או יותר קבוע למקור ה-WIFI, כאשר בכל מרחק מבין כמה מרחקים שנקבעו מראש, לוחצים על כפתור על מנת לדעת לשייך את המדידות שמתקבלות למרחק הנוכחי (יתכנו שגיאות מדידה, אך זו סוג של דרך לנסות העריך).

בבדיקת הנתונים השתמשנו בכמה אופציות לטיפול ברעשים, מתוכן ממוצע, חציון, ממוצע נע, [exponential smoothing](https://en.wikipedia.org/wiki/Exponential_smoothing), ו-Kalman filter, חלקם עם כמה פרמטרים שונים. ניתן לראות את התוצאות והגרפים שהתקבלו ב[אקסל](https://github.com/pliat963/Treasure-Hunt-Red/blob/main/wifi%20experiments/data/check%20options%20to%20deal%20with%20noise%20-%20with%20dists.xlsx), ובגרף המצורף כאן ניתן להציג רק חלק מהעקומות בעזרת הכפתור: . שמופיע כשלוחצים על הגרף.

השתמשנו במדידות אלה על מנת לחשב את המרחק, יחד עם פרמטרים n שנקבע במדידה נפרדת עי ממוצע של שלושה n-ים שהתקבלו ממרחקים 20 80 100 ס"מ עבור d0=50, ואילו P(d0) (שנסמן ב-B כעת) חושב כממוצע מניסוי זה (נבחרו ע"י ניסוי וטעיה, כי זה מה שהקטין את השגיאה - ערכים אחרים יכולים להוביל לשגיאות של יותר מ-50%).

התוצאות שהתקבלו – מכל אופציה לטיפול ברעש, בחרנו את מה שצמצם את השגיאה, כך שנותרנו עם האפשרויות (שצריך להמשיך לבדוק איך מתנהגות במציאות כדי לבחור):

1. ממוצע (וחציון) של כל 20 דגימות נתן שגיאה קטנה יחסית לאחרים, אבל היו רק מעט מאוד מדידות שנבדקו והושוו למרחק בפועל.
2. exponential smoothing () עבור נתן תוצאות טובות יחסית לאחרים.
3. גם ממוצע נע של 20 דגימות *נראה עם שגיאה קטנה יחסית (לאחרים).*
4. *בדיקה של Kalman filter.*

*חישובי המרחקים בעזרת האפשרויות שנבחרו:*

ממדידות שביצענו על מנת לחשב את הפרמטרים n ו-B, גילינו שקשה למצוא אותם, הערכים שקיבלנו משתנים אפילו באותה הסביבה, בין מדידות שונות, וזה משפיע על התוצאות. לכן אולי כדאי להוסיף מדידה שלהם בתחילת המשחק בתור ["כיול"](https://github.com/pliat963/Treasure-Hunt-Red/tree/main/wifi%20experiments/code/find_n_and_calculate_distance%20(calibration)) הכולל 3 מדידות ממרחקים שונים ולא גדולים כדי שיהיה ניתן לבצע בתחילת המשחק.

**ניסוי "חיפוש מטמון" בשטח:**

1. ממוצע של 20 דגימות בכל פעם:

גילינו שבחירת n ("כיול") טובה יותר עבור מרחקים של 50,80,100 ס"מ, מבין המרחקים שניסינו. ה-n במקרה זה: 2.27, עבור מרחק: d0=0.8.

עדיין – תוצאות לא נכונות ומרחק לא מדויק. סיבובים קטנים של הבקר במקום יכולים לשנות מאוד את המרחק, מרגיש קצת איטי (עובר זמן בין הדפסות של המרחק) והתוצאות בין מדידה למדידה יכולות לקפוץ כי מתייחסים לממוצע של מדידות נפרדות בכל הצגה של המרחק.

1. שימוש בKalman filter- - עדיין יש קפיצות לפעמים, והמרחק לא מדויק כל כך.

יתכן שהבעיה של המרחק קשורה לנוסחה שבחרנו ולקביעת הפרמטרים שלה. נראה שהנוסחה יכולה להיות נכונה בתחום קטן מאוד של מרחקים (יחסית קרובים), ולאחר מכן המרחק המתקבל ממנה מתחיל לגדול בקצב ממש מהיר, או לחלופין המרחק המתקבל יכול להיות קטן מאוד כל הזמן, בלי להשתנות הרבה ובלי לייצג נכון את המרחק כמעט בכלל. נראה שההתנהגויות האלה תלויות בפרמטרים שבוחרים, גם אם מודדים אותם ממש בתחילת המשחק ("כיול").

מסקנות: להמשיך לחפש.

חיפוש באינטרנט הוביל למציאת נוסחה נוספת <https://stackoverflow.com/questions/11217674/how-to-calculate-distance-from-wifi-router-using-signal-strength> המשמשת למדידת המרחק. לפי [זה](https://journals.sagepub.com/doi/full/10.1155/2014/371350) (נוסחה (1) המופיעה שם), היא (בניגוד לנוסחה שניסינו עד כה) משמשת לחישוב מרחק בתנאים אידיאליים (למשל ריק אינסופי סביב האנטנה, בלי מכשולים) ולכן אם תהיה בשימוש כאן, נראה שעדיף לבחור שטח פתוח עד כמה שאפשר.

בנוסף, מה שחשוב למציאת המטמון במשחק זה המונוטוניות של המרחק המחושב כתלות במרחק בפועל, ולא דווקא המרחק עצמו (כדי לדעת אם מתקרבים או מתרחקים). בגרף האחרון רואים שKalman ו-exponential smoothing נראים יחסית מונוטוניים וזו סיבה להמשיך איתם בבדיקות הבאות (למרות שהנוסחה לחישוב אחרת).

[**ניסויים עם הנוסחה ה"חדשה**":](https://github.com/pliat963/Treasure-Hunt-Red/tree/main/wifi%20experiments/code/another_formula_check_options) כעת, לפחות עד מרחק של כ-10 מטרים, אם פונים אל המטמון בקו יחסית ישר, המרחק נראה די קרוב למציאות (כרגע). עם סיבובים ותזוזות קטנות מהקו הישר זה, המרחק יכול להשתנות מאוד, אך ננסה עם סינון:

1. ניסיון: הנוסחה החדשה, עם Kalman. שינויים די מהירים, אך יחסית נראה בסדר. ניסיון להציג בכל פעם ממוצע של מספר **קטן** של מדידות (כלומר להוסיף גם ממוצע) עזר. יתכן שזו האופציה שתיבחר.
2. exponential smoothing עם אלפא = 0.1 והנוסחה החדשה – יש קפיצות במרחקים, מתעדכן פעם בכמה זמן, קצת יותר איטי, ועדיין שינויים די מהירים כשהמרחק בפועל משתנה לא כ"כ מהר.
3. ממוצע נע של 20 הדגימות האחרונות – שינויים מהירים.

וקצת ניסיונות (שלא נבחרו) לשילובים בניהם...

בינתיים נבחרה אופציה 3.

אחרי בדיקות נוספות בזמנים ומקומות אחרים, המרחק (גם עם הנוסחה החדשה) כבר לא נראה כ"כ נכון, אבל יחסית מונוטוני כשמתקרבים בקו ישר למטמון, וכשלא מכוונים את המחפש לעבר קיר או משהו דומה (כנראה בגלל החזרים של ה-Wifi).

ניסון לשיפור: הוספתparabolic reflector שהודפס בתלת מימד [(בתיקייה ASSETS)](https://github.com/pliat963/Treasure-Hunt-Red/tree/main/ASSETS) למחפש. זה אכן גרם למרחק להיות קטן יותר כשמצביעים לכוון של המטמון, מאשר לכיוונים אחרים (במקרים שבהם זה לא בהכרח היה כך), ובכך גרם ל"מרחק" הכתוב להיות אינדיקציה גם עבור הכיוון.