



אלגוריתמי ניווט ושיערוך מיקום

מחברת מתוך ההרצאות של ד"ר רועי יוזביץ
אוניברסיטת אריאל 2022

אלגוריתמי ניווט ושיערוך מיקום - מחברת הרצאות

נערך ונכתב על-ידי דור עזריה

2022

ספר זה לא נבדק על ידי מרצה, יתכן שימצאו טעויות.

לסיכומים נוספים שלי במדעי המחשב ומתמטיקה:

<https://dorazaria.github.io/>

מוזמנים לעקוב אחרי 😊 :

 LinkedIn: <https://www.linkedin.com/in/dor-azaria/>

 GitHub: <https://github.com/DorAzaria>

תוכן עניינים

3	מבוא הסתברותי
6	לוקליזציה ומסנן היסטוגרמה
13	מסנן קלמן במימד אחד
27	מסנן קלמן בשני מימדים
40	מסנן קלמן מורחב
45	לינאריזציה במסנן קלמן מורחב
48	היתוך חיישנים במסנן קלמן מורחב
54	מסנן חלקיקים - Particle Filter
64	בעיות חיפוש
68	אלגוריתם PID
75	ניווט בתוך מבנים

מבוא הסתברותי

אקסיומות ההסתברות

נגדיר Ω מרחב המדגם שלנו.

נגדיר A, B מאורעות אז:

$$1. \Pr(A) \geq 0$$

$$2. \Pr(\Omega) = 1$$

$$3. \Pr(A \cup B) = \Pr(A) + \Pr(B) \text{ אם } A \cap B = \emptyset$$

$$4. \Pr(A) = \frac{|A|}{|\Omega|}$$

שאלה

נתונה חבילת קלפים רגילה (52 קלפים) ומעורבת המכילה גם קלף ג'וקר אחד. (סה"כ 53 קלפים). משכתם ארבעה קלפים בזה אחר זה (בלי להחזיר לחבילה). מה ההסתברות שבאותם ארבעה קלפים יש קלף אחד מכל סוג (לב, תלתן, עלה ויהלום - סה"כ יש 13 מכל סוג).

פתרון

$$\frac{52}{53} \cdot \frac{39}{52} \cdot \frac{26}{51} \cdot \frac{13}{50} = 0.0975$$

הסתברות מותנית

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

שאלה

מטילים קובייה הוגנת בת 6 פאות, בהינתן שהתוצאה היא זוגית, מה ההסתברות שקיבלנו את המספר 2?

פתרון

נגדיר את המאורע B להיות המאורע שקיבלנו תוצאה זוגית.

נגדיר את המאורע A שקיבלנו את התוצאה 2.

מאחר וההסתברות אחידה כי הקובייה הוגנת אז נעזר בחישוב גודל קבוצה:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{|\{2\} \cap \{2,4,6\}|}{|\{2,4,6\}|} = \frac{|\{2\}|}{|\{2,4,6\}|} = \frac{1}{3}$$

הסתברות שלמה

$$\Pr(A) = \sum_{B_i \in B} \Pr(A|B_i) \cdot \Pr(B_i)$$

שאלה

מטילים מטבע, אם התוצאה יוצאת H אז עוצרים, אם קיבלנו T מטילים שוב ואז עוצרים. מה ההסתברות שההטלה הסופית היא T .

פתרון

התוצאה הסופית יכולה להיות עד 2 הטלות, כי אם H אז סיימנו כבר בראשון, ואם T נטיל שוב ואז נעצור. לכן נרצה לדעת מה ההסתברות לקבל T בהטלה השנייה. נגיד את המאורע T_2 שקיבלנו T בהטלה השנייה. נעזר בהסתברות השלמה:

$$\begin{aligned} \Pr(T_2) &= \Pr(T_2|H_1) \cdot \Pr(H_1) + \Pr(T_2|T_1) \cdot \Pr(T_1) = \\ &= 0 \cdot \frac{1}{2} + \frac{\Pr(T_2 \cap T_1)}{\Pr(T_1)} \cdot \Pr(T_1) = \\ &= 0 + \Pr(\{(T, T)\} \cap \{(T, T), (T, H)\}) = \Pr(\{(T, T)\}) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \end{aligned}$$

הסתברות בייס

נסמן $H = Hypothesis$ להיות ה-"השערה" ונסמן $E = Evidence$ להיות ה-"מציאות".

$$\Pr(H | E) = \frac{\Pr(E|H) \cdot \Pr(H)}{\Pr(E)}$$

שאלה

בדיקה מדווחת על קיום מחלה מסוימת כאשר אחוז הדיוק הוא 99%.
 • מתוך 100 חולים שיבדקו, 99 ידווחו כחולים ו-1 כבריא (בטעות).
 • מתוך 100 בריאים שיבדקו, 99 ידווחו כבריאים ו-1 כחולה (בטעות).
 בוחרים באקראי אדם מהאוכלוסייה ולפי הבדיקה יצא כי הוא חולה. מה הסיכוי שהוא באמת חולה בהינתן שהמחלה נפוצה באוכלוסייה ביחס של $\frac{1}{10,000}$.

פתרון

נסמן H אדם חולה ו- E בדיקה חיובית. נפעל לפי חוק בייס.

- $\Pr(E|H) = \frac{99}{100}$ כי זה ההסתברות שהבדיקה חיובית בהינתן החולה (לפי הנתון שלנו).
- $\Pr(H) = \frac{1}{10,000}$ זה ההסתברות שהאדם חולה לפי התפוצה באוכלוסייה.

- נרצה לחשב את $\Pr(E)$ כלומר שהבדיקה חיובית לפי הסתברות השלמה:

$$\begin{aligned}\Pr(E) &= \Pr(E|Sick) \Pr(Sick) + \Pr(E|Healthy) \Pr(Healthy) = \\ &= \frac{99}{100} \cdot \frac{1}{10,000} + \frac{1}{99} \cdot \left(1 - \frac{1}{10,000}\right)\end{aligned}$$

- ולפי בייס נקבל:

$$\Pr(H | E) = \frac{\Pr(E|H) \cdot \Pr(H)}{\Pr(E)} = \frac{\frac{99}{100} \cdot \frac{1}{10,000}}{\frac{99}{100} \cdot \frac{1}{10,000} + \frac{1}{99} \cdot \left(1 - \frac{1}{10,000}\right)} \approx 1\%$$

לוקליזציה ומסנן היסטוגרמה

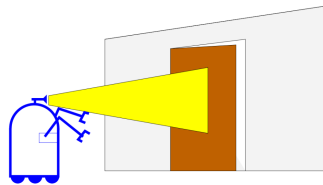
הגדרות

- לוקליזציה גלובלית - היכולת לדעת איפה אני נמצא כאשר מתחילים מחוסר ודאות מוחלט.
- לוקליזציה לוקאלית - היכולת לדעת איפה אני נמצא עכשיו בהינתן שידעתי את מיקומי הקודם.
- מסנן היסטוגרמה - חלוקת העולם ל-Grid (משבצות), כלומר עולם בדיד.
- הסתברות אפרורית - הסתברות שנמצאת לפני שאנחנו יודעים על משהו.
- הסתברות אפוסטריורית - האינפורמציה שקיבלנו מהחיישן בהסתברות מסוימת.

בעיות לוקליזציה

נציג דוגמה עבור בעיית לוקליזציה.

- יש לנו רובוט נייד שמנסה ללמוד לוקליזציה, הוא נמצא במסדרון עם 3 דלתות והוא יודע לסרוק אותם.
- הרובוט לא יודע איפה הוא נמצא ואיזה דלת הוא סורק, הוא רק יודע לסרוק דלתות.

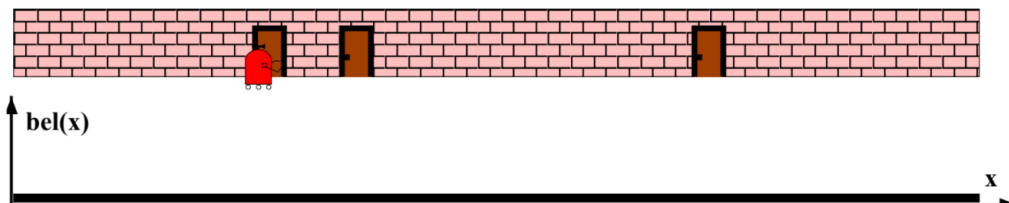


נציג בשלבים את הבעיה לפי הדוגמה:

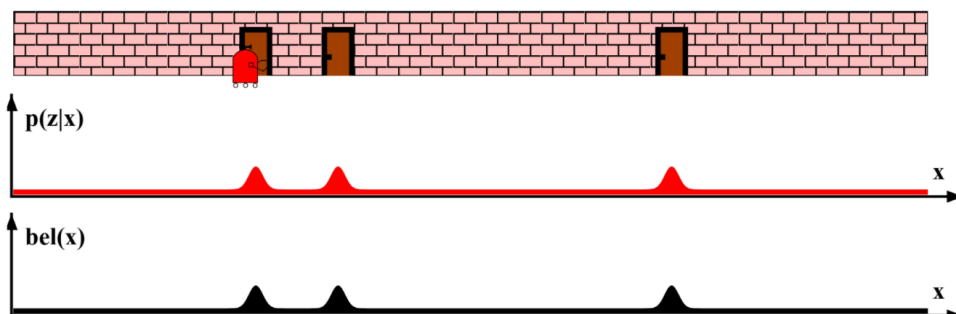
- ציר ה-x הוא מיקום ה-x שהרובוט יכול למדוד.
 - פונקציית האמונה** $belief$ ונסמן $bel(x)$ היא איפה הרובוט מאמין שהוא נמצא.
- כלומר, היא ההערכה הנוכחית של הרובוט על המיקום ומיוצגת ע"י הפונקציה שהיא ההסתברות על פני כל המרחב של כל המיקומים.

1. הרובוט כרגע נמצא במצב של לוקליזציה גלובלית.

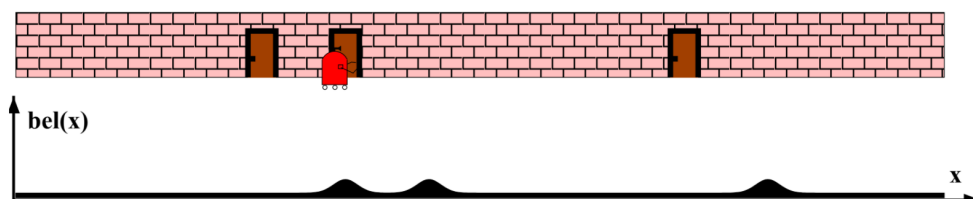
אם הרובוט בחוסר וודאות מוחלט לגבי מיקומו, אז המקום שהוא נמצא מתפלג בהסתברות אחידה על כל המרחב.



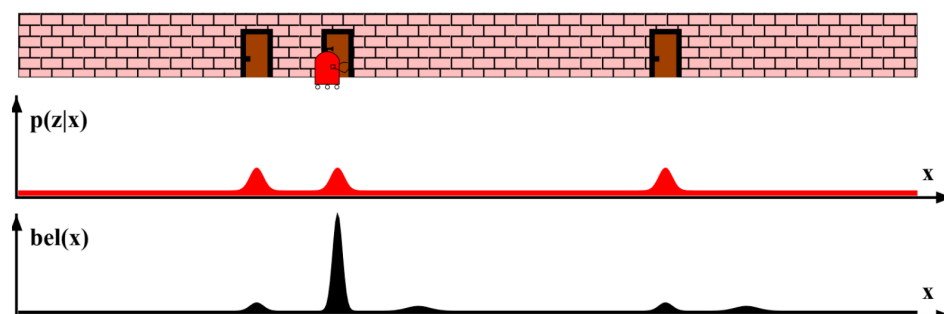
2. נניח שהרובוט לוקח מדידה ראשונה בעזרת החיישן שלו, כל מדידה היא סריקה לאורך המסדרון. הרובוט מגלה כי הוא נמצא מול דלת כלשהי. פונקציה האמונה $bel(x)$ מציבה הסתברות גבוהה במקומות ליד דלתות והסתברות נמוכה במקומות אחרים. שימו לב שלתפוצה זו יש שלושה פסגות, שכל אחת מהן מתאימה לאחת הדלתות (שלא ניתן להבחין בהן) בסביבה.



3. כעת נניח שהרובוט יזוז 10 מטר, בכל תזוזה אנחנו מורידים אינפורמציה (כי זה כמו ללכת בעיניים עצומות) בהנחה שהרובוט זז כמצוין, האמונה מוסטת לכיוון התנועה. אפשר לראות את אי הוודאות המובנית בתנועת הרובוט, הוודאות יורדת בתנועה.

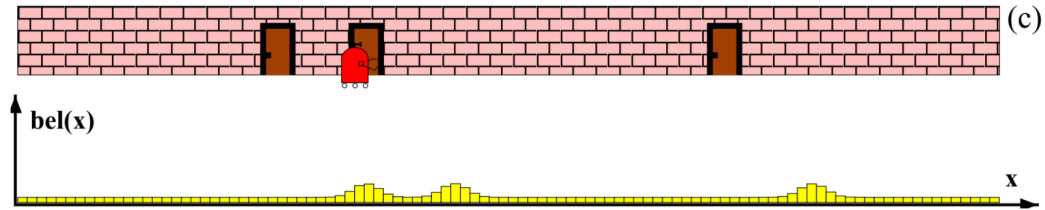


4. לאחר כ-10 מטרים הרובוט נפגש שוב בדלת כלשהי. הציור מתאר את האמונה לאחר התבוננות בדלת אחרת. תצפית זו מובילה את האלגוריתם שלנו למקם את רוב מסת ההסתברות במיקום ליד אחת הדלתות, והרובוט בטוח כעת היכן הוא נמצא.



מסנן היסטוגרמה

מסנן היסטוגרמה מחלקת את המרחב למשבצות. המסנן נותן פתרון לבעיות לוקליזציה גלובלית, למשל לפי הדוגמה הקודמת:



נעזר במקרים קטנים יותר להבנה:

נניח והעולם שלנו מחולק ל-5 משבצות, אז ההסתברות להיות בכל תא היא $\frac{1}{5}$ כלומר:

1	2	3	4	5
$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$

כעת נניח ולרובוט שלנו יש חיישן שמזהה ירוק ב-100%, וקיבלנו חייווי לירוק אז ההסתברות לכל משבצת היא:

1	2	3	4	5
0	$\frac{1}{2}$	$\frac{1}{2}$	0	0

אלגוריתם ניווט בייסאני

כעת נניח שהחיישן מזהה ירוק ב- $\frac{3}{4}$ ואדום ב- $\frac{1}{4}$, מה נעשה עכשיו? נשתמש בחישוב על פי בייס.

$$\Pr(H | E) = \frac{\Pr(E|H) \cdot \Pr(H)}{\Pr(E)}$$

נסמן H ההשערה שהרובוט בתא ירוק ו- E שקיבלנו מהחיישן שאנחנו בתא ירוק.

- $\Pr(E|H) = 0.6$ זה $\frac{3}{5}$.
- $\Pr(H) = 0.4$ כי יש 2 משבצות ירוקות מתוך ה-5, כלומר ההסתברות להיות בכל תא זה 0.2 בלי קשר לצבע, ובגלל שיש 2 תאים ירוקים אז $0.2 + 0.2 = 0.4$.
- נרצה לחשב את $\Pr(E)$ לפי ההסתברות השלמה:

$$\Pr(E) = 0.6 \cdot 0.4 + 0.2 \cdot 0.6 = 0.36$$

- סה"כ קיבלנו:

$$\Pr(H | E) = \frac{\Pr(E|H) \cdot \Pr(H)}{\Pr(E)} = \frac{0.6 \cdot 0.4}{0.36} = \frac{2}{3}$$

- ומאחר ואנו רוצים לדעת את ההסתברות להיות בתא ירוק יחיד אז $\frac{\frac{2}{3}}{2} = \frac{1}{3}$.

פונקציית התזוזה ההסתברותית

ניתן לשערך בצורה טובה איפה הרובוט נמצא, אבל בעולם האמיתי כאשר הרובוט זז, הוא לא יודע בוודאות אם הוא זז בכלל וכמה הוא זז לכן הרובוט ירצה לזוז בעולם הסתברותי. אם נחזור לדוגמה של המשבצות, כאשר הרובוט יזוז על המשבצות הרבה זמן, יכולות להתחיל להיות טעויות קטנות בזמן התנועה.

נציג תחילה שני מושגים:

1. Overshoot - הרובוט זז יותר מהצפוי.
2. Undershoot - הרובוט זז פחות מהצפוי.

נסביר בדוגמה, נניח שהרובוט מתחיל בתא ה-1 עם 100% וודאות לכן נסמן את ההסתברות:

1	2	3	4	5
1.0	0.0	0.0	0.0	0.0

כעת, נרצה לזוז בין המשבצות אבל בהסתברות מסוימת. נשים לב שיש הבדל בין לזוז בבת אחת 2 משבצות לבין תזוזה של פעמיים משבצת אחת - זה מכיוון שבכל תזוזה אנחנו מכניסים עוד אי-ודאות למערכת.

- כעת, נניח שהרובוט עבר $x = 2$ צעדים, לכן:

$$\Pr(\text{Exactly}) = 0.8, \quad \Pr(\text{Overshoot}) = \Pr(\text{Undershoot}) = 0.1$$

כלומר נקבל:

1	2	3	4	5
0.0	0.1	0.8	0.1	0.0

כעת, נרצה למשל מתא 2 לקפוץ 2 צעדים לתא 4 כלומר נשאל איך יראו ה-Under ו-Over שוטים במקרה פרטי זה:

1	2	3	4	5
0.0	0.0	0.1	0.8	0.1

לכן ההסתברות לזוז לתא 3 מתא 2 יהיה $0.1 \cdot 0.1 = 0.01$ (נעזר ב-2 המערכים ממש מעלינו).
ההסתברות לזוז לתא 4 מתא 2 יהיה $0.1 \cdot 0.8 = 0.08$
וההסתברות לזוז לתא 5 מתא 2 יהיה $0.1 \cdot 0.1 = 0.01$

- נחשב את ההסתברות להגיע לכל תא בצעד השני:

מתא 4	מתא 3	מתא 2	תזוזה לתא / התא שהייתי
		$0.1 \cdot 0.1 = 0.01$	לתא 3
	$0.8 \cdot 0.1 = 0.08$	$0.1 \cdot 0.8 = 0.08$	לתא 4
$0.1 \cdot 0.1 = 0.01$	$0.8 \cdot 0.8 = 0.64$	$0.1 \cdot 0.1 = 0.01$	לתא 5
$0.1 \cdot 0.8 = 0.08$	$0.8 \cdot 0.1 = 0.08$		לתא 1
$0.1 \cdot 0.1 = 0.01$			לתא 2

אם נסכום כל שורה בטבלה נקבל:

1	2	3	4	5
0.16	0.01	0.01	0.16	0.66

הסכימה הכללית הזאת שעשינו אומרת לנו בעצם מה הסיכוי האמיתי להיות בכל משבצת. למשל, הסיכוי האמיתי והכולל שלנו להיות במשבצת מספר (1) הוא 0.16.

אנטרופיה ואינטגרציה

האנטרופיה מגדירה עד כמה המשתנה אקראי.

האנטרופיה מוגדרת כך:

$$H(x) = - \sum_x \Pr(x) \cdot \log(\Pr(x))$$

למשל, אם יש לנו מטבע מזויף (לא הוגן) בו שני הצדדים הם H אז ההסתברות לקבל H תהיה 1 ול- T תהיה 0.

- ככל שהאנטרופיה גדולה יותר יש יותר אקראיות.
- וודאות גבוהה $\Leftarrow$ אנטרופיה נמוכה.
- וודאות נמוכה $\Leftarrow$ יש יותר אי וודאות $\Leftarrow$ האנטרופיה גבוהה.

סיכום

אלגוריתם ניווט בייסאני

כזכור, חוק בייס מחשב את ההיפותזה בהינתן ראיה (או מדידה).
 מהי ההיפותזה בעולמות הניווט? היכן נמצא הרכב (רובוט) שלי?
 השאלה היא באיזו משבצת אני נמצא. הראיה או המדידה הן המידע שאנחנו מקבלים מהחיישנים.
 אם החיישן "הרגיש" אדום, הסיכוי שאני במשבצת אדומה גבוה יותר.
 השאלה שחוק בייס פותר היא מהי הדרך לשנות את האמונה הראשונית $\Pr(H)$ או את האמונה
 האפריורית, לפי המדידה שהתקבלה $\Pr(H|E)$.

מסנן היסטוגרמה

- כאשר מוסיפים לו ממדים, הבעיה נהיית מורכבת חישובית בצורה מעריכית (אקספוננציאלית בלעז).
- מסנן היסטוגרמה הוא בגדול מסנן בדיד (Discrete). בדוגמאות שראינו, המשמעות היא שהרובוט יכול להיות רק באחת מתוך המשבצות האפשריות. יתרה מזאת, אם גודל המשבצת הוא 10 מ"ר, אזי ההסתברות להיות בכל נקודה בתוך השטח היא שווה.

אנטרופיה

- חשוב לזכור שפונקציית החישה $bel(x)$ מעלה תמיד את הדיוק (מורידה את האנטרופיה), ופונקציית התזוזה מורידה את הדיוק (ומעלה את האנטרופיה).

הנחת מרקוב

הנחת מרקוב מניחה שנתוני עבר ועתיד הם בלתי תלויים אם יודעים את המצב הנוכחי.
 כלומר, אם ההווה נתון לנו, אז העבר והעתיד בלתי תלויים. - זה נושא קצת פילוסופי.

נסמן:

Z - מדידה

U - וקטור התנועה

X - וקטור המצב

נרצה לדעת לשערך מהו וקטור המצב X_t .

$$\Pr(X_t | X_0, X_1, \dots, X_{t-1}, Z_0, Z_1, \dots, Z_t, U_0, \dots, U_{t-1})$$

כלומר בהינתן כל המידע שיש לנו לאורך כל ההיסטוריה שלנו החל מ-0 עד לנקודה $t - 1$, אז מה יהיה המצב שלנו בנקודה X_t ?

מה הבעיה? - אם נרצה לדעת מהו X_t בהינתן כל המידע שקרה לפניו אז אנחנו בבעיה קשה כי אי אפשר להכיל את כל המידע הזה. - יכול להיות כמויות מטורפות של נתונים.

פתרון - זה טריק בעייתי ופילוסופי - נציג כך:

$$\Pr(X_t | X_0, X_1, \dots, X_{t-1}, Z_0, Z_1, \dots, Z_t, U_0, \dots, U_{t-1}) = \Pr(X_t | X_{t-1}, Z_t, U_{t-1})$$

כלומר בהינתן המדידות ההכי אחרונות, נוכל לקבל את המדידה הנוכחית. ההבדל הוא שכאן החישוב הוא הרבה יותר קטן ממה שהיה לנו לפני. האינדיקציה הכי טובה לקבל מה יהיה בעתיד זה לפי מה שקורה עכשיו, זה לא משנה אם נתן לו את כל ההיסטוריה כי כל המידע הזה מוכל בתוך X_{t-1} . אבל בחיים זה לא באמת נכון, לכן הנחת מרקוב היא רק הנחה ולא וודאי.

מסנן קלמן במימד אחד

מסנן קלמן מאפשר לפתור את בעיית המיקום בצורה רציפה ולקבל הערכת מיקום שונה עבור כל נקודה.

הגדרה

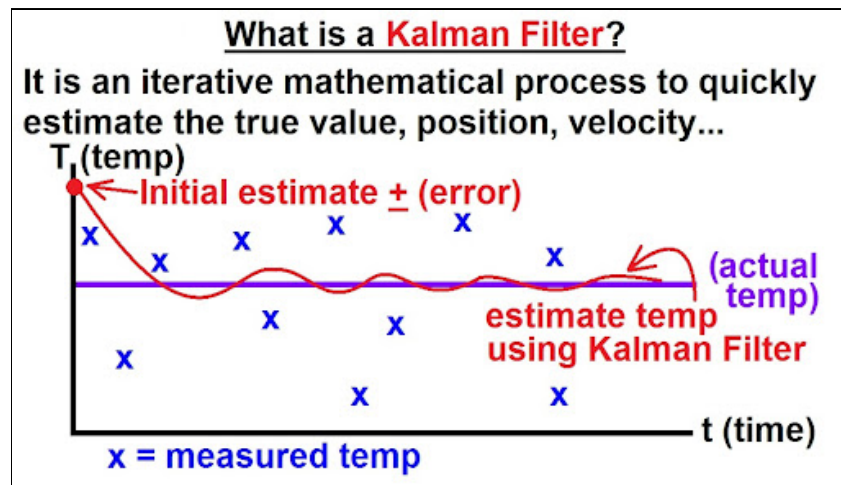
- מסנן קלמן זה תהליך מתמטי איטרטיבי שמשתמש בסט של משוואות ונתונים שמגיעים ברציפות מהחיישנים כדי לשערך במהירות את הערך האמיתי של אובייקט, כאשר הערכים הנמדדים מכילים חוסר וודאות, רעש ואקראיות.
- ניתן לחשוב על מסנן קלמן כעל ממוצע משוקלל בין המדידה הנוכחית מהחיישן ובין הידע הקודם שלנו לגבי מצב המערכת.
- הרכיב שקובע את המשקל של כל אחד מהרכיבים הוא הגבר קלמן.

יתרונות של מסנן קלמן

1. **שיערוך עקיף** - שיערוך על פי כמה פרמטרים שונים כמו מהירות ומיקום.
2. **היתוך חיישנים** - תהליך מתמטי של חיבור מידע מחיישנים שונים.

הגדרות נוספות

- תהליך מתמטי איטרטיבי - שימוש חוזר בסט משוואות זהה, עם נתונים שונים.
- אצלנו התפלגות שחישבנו בזמן $t - 1$ הופכת לבסיס החישוב עבור זמן t .
- שיערוך מהיר של הערך האמיתי (וקטור המצב), מיקום ומהירות.
- מדידות המכילות חוסר וודאות - החיישנים מכילים רעש.



הגבר קלמן (Kalman Gain)

הגבר קלמן או KG עוזר בממוצע המשוקלל בין השערוך בזמן הקודם ($t - 1$) ובין המדידה בזמן הנוכחי (t). הבעיה שיש לנו חוסר וודאות מצד השערוך ומצד המדידה:

- כשאני רוצה לנחש איפה הרובוט שלי נמצא, אז בשערוך (בניחוש) יש חוסר וודאות שנקרא $Eest$.
- כשאני מודד איפה אני נמצא עם החיישנים יש לי גם חוסר וודאות במדידה $Emeas$.
- שני הפרמטרים האלה נכנסים במשוואה שנקראת קלמן $Kalman Gain$.

נסמן:

- $Eest$ - השגיאה בשערוך (חוסר וודאות בשיערוך).
- $Emeas$ - השגיאה במדידה (חוסר וודאות במדידה).
- $MEAS$ - המדידה.
- EST - השערוך.
- להלן המשוואות לפי שלבים:

$$1. \quad KG_t = \frac{Eest_{t-1}}{Eest_{t-1} + Emeas_t} \quad \text{כאשר } 0 \leq KG \leq 1$$

$$2. \quad EST_t = EST_{t-1} + KG_t \cdot [MEAS_t - EST_{t-1}]$$

$$3. \quad Eest_t = [1 - KG_t] \cdot [Eest_{t-1}]$$

מסקנות:

- ככל שה- KG יותר קטן, כך שערוך המצב הקודם מדויק יותר ביחס למדידה הרועשת.
- ככל שה- KG גדול יותר, כך המדידה הנוכחית מדויקת יותר על פני השערוך הקודם.
- הרעיון מאחורי מסנן קלמן הוא מעבר על שלבים 1-3 בצורה איטרטיבית, ובכל שלב, להקטין את ה- KG להגדיל בכך את האמונה שלנו בשערוך הנוכחי.
- המשמעות היא שניתן לקבל מדידות רועשות ובכל זאת לנצל אותן על מנת לדייק בניחוש.

ניתוח הנוסחאות

$$\text{עבור הנוסחה } KG_t = \frac{Eest_{t-1}}{Eest_{t-1} + Emeas_t} \quad \text{כאשר } 0 \leq KG \leq 1$$

- $KG \rightarrow 1$ כאשר $KG \approx \frac{Eest}{Eest}$
 - יש לי יותר וודאות במדידה $\Leftarrow$ המדידה דיי מדויקת.
 - יש לי הרבה חוסר וודאות בשערוך $\Leftarrow$ מצב לא יציב.

$$\bullet \quad KG \rightarrow 0 \quad \text{כאשר } Eest_t < Emeas_t$$

- המדידה רועשת.
- שיערוך במצב לא יציב.

$$EST_t = EST_{t-1} + KG_t \cdot [MEAS_t - EST_{t-1}]$$

עבור הנוסחה

- אם $KG \rightarrow 0$ נקבל ש: $EST_t = EST_{t-1}$
 - כלומר, השיערוך במצב מדויק.
- אם $KG \rightarrow 1$ נקבל $EST_t = MEAS_t$
 - כלומר, השגיאה כמעט אפסית, ועדיף להתייחס רק אליה (כי השיערוך דייק במדידה).
- עבור $KG_t \cdot [MEAS_t - EST_{t-1}]$ בעצם מחשב לנו את ההפרש בין המדידה לשיערוך.

תרגיל

חשבו בהתאם לנתוני המדידה (MEAS) ולנתוני השגיאה במדידה (Emeas) את השיערוך (EST), את השגיאה בשיערוך (Eest), את הגבר קלמן (KG) ואת השגיאה הקודמת בשיערוך (Eest t-1).
(כל מה שכתוב בכתום היה כתוב מראש והיה צריך למלא את שאר הנתונים על סמך הקיימים):

	MEAS	Emeas	Est	Eest	KG	Eest _{t-1}
t - 1			68			2
t	75	4	70.33	1.33	0.33	
t + 1	71	4	70.50	1.00	0.25	
t + 2	70	4	70.4	0.80	0.20	
t + 3	74	4	71	0.66	0.17	

$$kg_t = \frac{E_{est_{t-1}}}{E_{mes} + E_{est_{t-1}}} = \frac{2}{4+2} = 0.33$$

$$Est_t = Est_{t-1} + kg (Mes - Est_{t-1}) = 68 + 0.33 (75 - 68) = 70.33$$

$$E_{est_t} = [1 - kg] (E_{est_{t-1}}) = 0.66 * 2 = 1.33$$

$$kg_{t+1} = \frac{1.33}{4 + 1.33} = 0.25$$

$$Est_{t+1} = \underbrace{70.33}_{Est_t} + \underbrace{0.25}_{kg} \left(\underbrace{71}_{Mes} - \underbrace{70.33}_{Est_t} \right) = 70.5$$

$$E_{est_{t+1}} = \underbrace{0.75}_{1-kg} * \underbrace{1.33}_{E_{est_t}} = 1$$

$$kg_{t+2} = \frac{1}{4 + 1} = 0.2$$

$$Est_{t+2} = \underbrace{70.5}_{Est_{t+1}} + \underbrace{0.2}_{kg} \left(\underbrace{70}_{Mes} - \underbrace{70.5}_{Est_{t+1}} \right) = 70.4$$

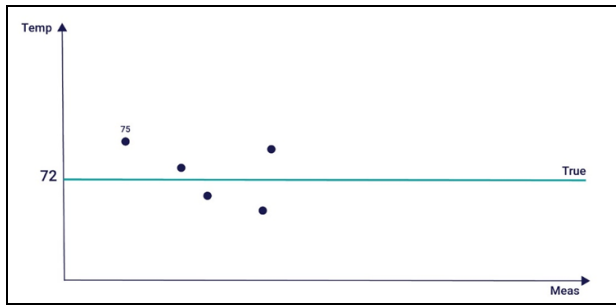
$$E_{est_{t+2}} = \underbrace{0.8}_{1-kg} * \underbrace{1}_{E_{est_{t+1}}} = 0.8$$

$$kg_{t+3} = \frac{0.8}{4 + 0.8} = 0.17$$

$$Est_{t+3} = \underbrace{70.4}_{Est_{t+2}} + \underbrace{0.17}_{kg} \left(\underbrace{71}_{Mes} - \underbrace{70.4}_{Est_{t+2}} \right) = 71$$

$$E_{est_{t+3}} = \underbrace{0.83}_{1-kg} * \underbrace{0.8}_{E_{est_{t+2}}} = 0.66$$

תרגיל



הטמפרטורה האמיתית היא 72 מעלות. הניחוש הראשוני שלנו הוא 68 מעלות, והטעות הראשונית (Eest) היא 2 מעלות. המדידה הראשונית היא 75 מעלות עם טעות של 4 מעלות לכל כיוון. הכניסו את הנתונים לסט המשוואות וחשבו את ה- KG , EST_t ו- $Eest$ בטבלה.

נחשב:

	$Meas$	E_{mes}	kg	E_{est}	Est
t	75	4	$\frac{E_{est}}{4+2} = 0.33$	$\left[1 - \underbrace{0.33}_{kg}\right] (2) = 1.33$	$\underbrace{68}_{Est_{t-1}} + \underbrace{0.33}_{kg} \left(\underbrace{75}_{Meas} - \underbrace{68}_{Est_{t-1}} \right) = 70.33$

מאפיינים של מסנן קלמן

שני מאפיינים של מסנן קלמן אל מול מסנן היסטוגרמה.

1. Multimodal מול ה-Unimodal

המאפיין הראשון זה מה שנקרא **Multimodal**, שזה היה מסנן ההיסטוגרמה, מול **Unimodal**.

- ב-Multimodal יש לי כמה ניחושים שונים שאני יכול להחזיק אותם ואין לי עניין להעדיף ניחוש אחד על פני ניחוש אחר - למשל, אני נמצא בקומה מסוימת ואני יכול להיות בקומה 1, 2 או 3 ואני מחזיק בניחושים האלה.
- לעומת זאת ב-Unimodal יש לי רק ניחוש אחד עם חוסר ודאות - זה מה שאנחנו עושים במסנן קלמן, אנחנו מחזיקים בניחוש אחד ויחיד אבל הוא הניחוש הטוב ביותר.

2. Discrete מול ה-Continuous

המאפיין השני זה **Discrete** או בדיד שזה היה מסנן ההיסטוגרמה, אל מול **Continuous** או רציף, שעליו נדבר במסנן קלמן.

- ב-Discrete שזה מה שעשינו במסנן ההיסטוגרמה. כלומר אני עובר מכל התא לכל התא, הרעיון הזה נקרא Discrete, בדיד. יש לי מספר סופי של תאים וככל שהרזולוציה שלי גבוהה יותר, יש לי מספר גדול יותר.
- ב-Continuous שזה מה שנעשה במסנן קלמן, אני יכול להיות בכל נקודה ויש הבדל בין אם אני בנקודה הזאת ובין אם בנקודה הזאת.

לסיכום:

1. **מסנן קלמן הוא Unimodal ו-Continuous.**
2. **מסנן היסטוגרמה הוא Multimodal ו-Discrete.**

נקודה חשובה

על מנת לשפר את הדיוק במסנן היסטוגרמה, אני צריך שהגריד או שהרשת שלי תהיה כמה שיותר קטנה ולכן הרזולוציה תהיה כמה שיותר גדולה - הדבר הזה מצריך הרבה מאוד משאבים של זיכרון. בקלמן פילטר אין לי את הדבר הזה, זאת אומרת אני יכול לקחת את ההסתברות שלי ולהזיז אותה בדיוק באיזו נקודה שאני רוצה (כי העולם רציף).

ממוצע, שונות וסטיית התקן

- נוסחת ממוצע - $\mu = \frac{\sum_{a \in A} (a)}{|A|}$
- נוסחת סטיית תקן - $\sigma = \sqrt{\frac{\sum_{a \in A} (a - \mu)^2}{|A|}}$

תרגיל

להלן שלושה מקבצי נתונים A, B ו-C:

$$A = \{9, 10, 11, 7, 13\}, \quad B = \{10, 10, 10, 10, 10\}, \quad C = \{1, 1, 10, 19, 19\}$$

חשבו את הממוצע של כל אחד מהם.

פתרון

$$A = \frac{9+10+11+7+13}{5} = 10, \quad B = \frac{5 \cdot 10}{5} = 10, \quad C = \frac{2 \cdot 1 + 10 + 2 \cdot 19}{5} = 10$$

תרגיל

בהמשך לשאלה הקודמת, למי סטיית התקן הגדולה ביותר?

פתרון

$$A = \sqrt{\frac{(9-10)^2 + (10-10)^2 + (11-10)^2 + (7-10)^2 + (13-10)^2}{5}} = \sqrt{\frac{(-1)^2 + 0 + (1)^2 + (-3)^2 + (3)^2}{5}} = 2$$

$$B = \sqrt{\frac{(10-10)^2 + (10-10)^2 + (10-10)^2 + (10-10)^2 + (10-10)^2}{5}} = \sqrt{\frac{0}{5}} = 0$$

$$C = \sqrt{\frac{2 \cdot (1-10)^2 + (10-10)^2 + 2 \cdot (19-10)^2}{5}} = \sqrt{\frac{2 \cdot (-9)^2 + 0 + 2 \cdot (9)^2}{5}} = \frac{18\sqrt{5}}{5} \approx 8.0498$$

לכן ל-C יש את סטיית התקן הגדולה ביותר.

תרגיל

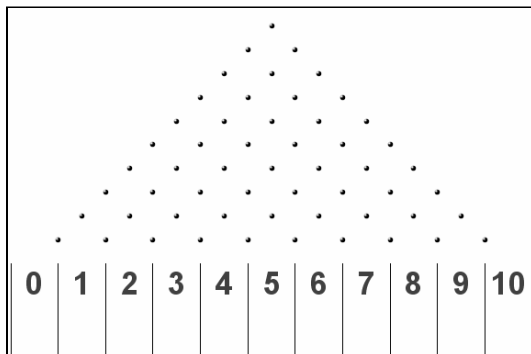
קיים מקבץ נתונים בעל ממוצע μ וסטיית תקן σ .

מה יהיה הממוצע החדש, אם לכל אחת מהנקודות ב-DATA יתווסף קבוע מסוים K?

פתרון - $\mu' = \mu + K$

התפלגות גאוסית

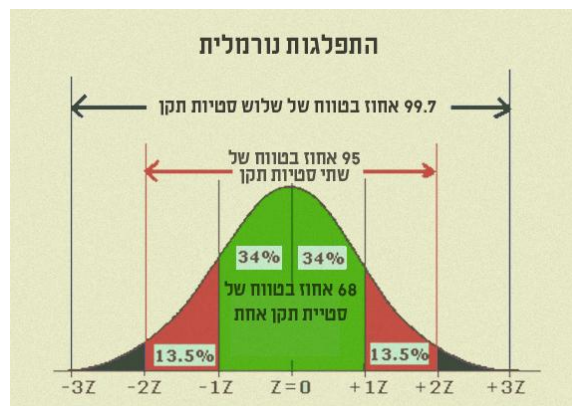
- מהי בעצם התפלגות היסטוגרמה? זו התפלגות אשר לאורך המרחב, יש בכל נקודה ערך אחר שונה. בהתפלגות היסטוגרמה אני יכול להכיל או להחזיק שני ניחושים שכל אחד מהם טוב באותה מידה.
- התפלגות גאוסית או התפלגות נורמלית מאופיינת בשני פרמטרים בלבד.
 - הפרמטר הראשון נקרא "הממוצע" או μ
 - הפרמטר השני נקרא "סיגמא" או סטיית התקן σ
- באמצעות שני הפרמטרים האלה, אנחנו יכולים לדעת את כל מה שאנחנו רוצים על התפלגות גאוסית.
 1. מה ההסתברות לקבל ערך מסוים?
 2. מה ההסתברות לקבל ערכים בין תחום כזה לתחום כזה? (בעזרת אינטגרל)
- במילים אחרות, הכל ניתן לחישוב באמצעות שני ערכים בלבד.
- הדרך המתמטית לטפל בהתפלגות גאוס היא באמצעות ה- **Probability Density Function**. או פונקציית צפיפות ההסתברות או ה- **PDF**.
- פונקציית ההתפלגות:



$$N(\mu_1, \sigma_1^2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \cdot e^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}}$$

חוק ה-99.7, 95, 68

ההתפלגות הסימטרית השכיחה ביותר נקראת התפלגות נורמלית (גאוס). הפונקציה שלה מתארת עבור כל ערך של המשתנה שאנו מעוניינים לדגום מה אחוז הפעמים שנדגום אותו, כלומר מה ההסתברות שהוא יתקבל. כל התפלגות נורמלית מתוארת על ידי שני משתנים – הממוצע והשונות. כפי שאפשר לראות באיור, ההתפלגות הנורמלית היא סימטרית ונראית קצת כמו פעמון, עם שיא אחד שמייצג את הממוצע (ששווה במקרה הזה לחציון). המשמעות היא שהכי סביר שערך אקראי יימצא בדיוק על הממוצע, וככל שמתרחקים ממנו ההסתברות יורדת.

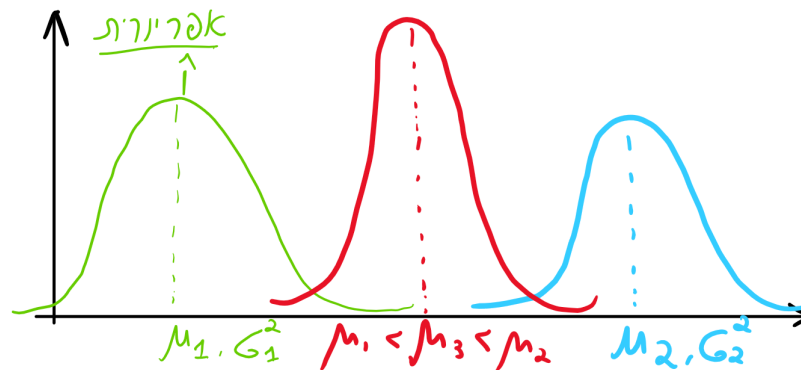


מכפלת גאוסיים

הניחוש שלנו למיקום הרכב (אפריורי / Prior) יש ממוצע μ_1 וגם שונות σ_1^2 .
 הסתברות המדידה למיקום הרכב יש ממוצע μ_2 וגם שונות σ_2^2 .
 הכפלה של שני גאוסיאנים יוצרת גאוסיאן שלישי אשר הממוצע שלו μ_3 נמצא בין שני הממוצעים המקוריים.
 התוצאה של הכפלת גאוסיינים היא שהשונות והממוצע יותר קטנה אחרי הכפלה.
 התהליך של הכפלה מעלה הדיוק ולכן מוריד חוסר וודאות.
 את ההכפלה נציג לפי הנוסחה הבאה:

$$\mu_3 = \frac{\mu_1 \cdot \sigma_2^2 + \mu_2 \cdot \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad \sigma_3^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}$$

עם זאת, הדבר המפתיע אשר עולה מהמשוואות הוא שסטיית התקן או חוסר הוודאות של הגאוסיאן החדש, קטנה מכל אחת מסטיות התקן המקוריות.
 ביישום התופעה הזו על הסתברות מיקומה של מכונת, אפשר להסיק שאם כל אחד משני הסנסורים (גאוסיאן) מתאר אמונה שונה שלנו במיקומה, אזי שילוב שתי האמונות האלה ייתן ניחוש שהוא ודאי יותר מכל אחד מהסנסורים בנפרד.
 באיור שלפניכם אפשר לראות את שני הגאוסיאנים המקוריים בצבעים **ירוק ותכלת**, ובצבע **אדום** - את הגאוסיאן הנוצר מהמכפלה שלהם.



לדוגמה עבור רכבים:

- בירוק - הניחוש שלנו למיקום הרכב (אפריורי / Prior).
- בכחול - הסתברות המדידה למיקום הרכב.
- עולה השאלה בשקלול הגאוסיינים המבטאים את מיקום הרכב, היכן אנחנו באמת? ולכן סביר שהמיקום שלנו הוא **ממוצע** כלשהו ביניהם וזה באדום (אפוסטרירי).

כלומר גם אם ב- σ_1^2, σ_2^2 יש חוסר וודאות - הכפלה של שניהם תתן לנו את σ_3^2 שתבטיח תוצאה מדויקת יותר!

מקרי קצה ונקודות חשובות

$$1. \sigma_3^2 \leq \sigma_1^2, \sigma_2^2$$

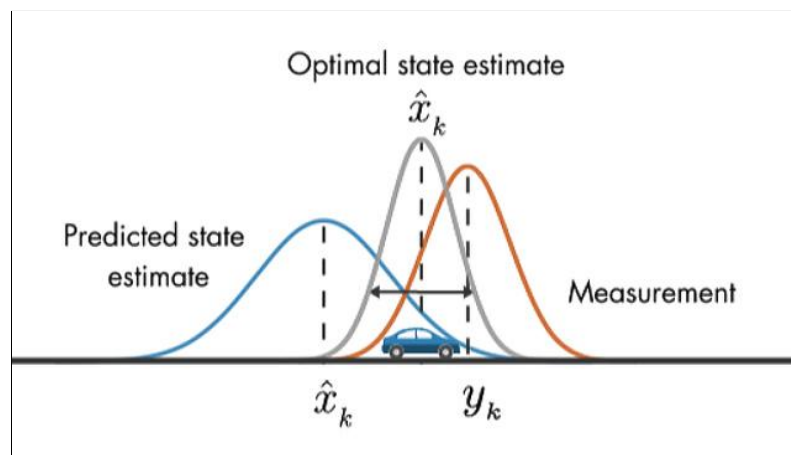
$$2. \text{אם } \sigma_1 = \sigma_2 \text{ אז } \sigma_3^2 = \frac{1}{2} \cdot \sigma_1^2$$

$$3. \text{אם } \sigma_1 \rightarrow 0, \text{ כלומר יש לי 0 חוסר ודאות אז זה אומר שאני סומך מאוד על } \mu_1 \text{ עד כדי שגם } \sigma_3^2 = 0$$

כלומר חוסר הוודאות של σ_3 היא גם 0 ומכאן גם $\mu_3 \rightarrow \mu_1$.

$$4. \text{אם } \sigma_1 \rightarrow \infty \text{ כלומר יש לי } \infty \text{ חוסר ודאות } \mu_1 \text{ עד כדי שעדיף לי להתעלם ממנו באופן מוחלט ולכן}$$

נעדיף לבחור ב- σ_2 כלומר $\sigma_3^2 \rightarrow \sigma_2^2$ וגם $\mu_3 \rightarrow \mu_2$ (כלומר נעדיף להתעסק עם מדידה חדשה).

**תרגיל**

נתונות שתי התפלגויות גאוסיות:

$$\text{לאחת יש ממוצע } \mu_1 = 10 \text{ ושונות } \sigma_1^2 = 8 \text{ ולאחרת יש ממוצע } \mu_2 = 13 \text{ ושונות } \sigma_2^2 = 2.$$

מה הממוצע והשונות החדשים של ההכפלה בין שני הגאוסיאנים הללו?

פתרון:

$$\sigma_3^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{1}{\frac{1}{8} + \frac{1}{2}} = 1.6$$

$$\mu_3 = \frac{\mu_1 \cdot \sigma_2^2 + \mu_2 \cdot \sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{10 \cdot 2 + 13 \cdot 8}{8 + 2} = \frac{20 + 104}{10} = 12.4$$

תרגיל

נתון רובוט אשר נמצא בעולם חד ממדי. מיקום הרובוט ניתן על ידי PDF גאוסית עם ממוצע של 7 מטרים ושונות של 5 מטרים בריבוע. ברגע מסוים, חיישן GPS מדווח שהרובוט נמצא דווקא במיקום 11 ולא 7. לחיישן ה-GPS יש חוסר דיוק מובנה המתאפיין בשונות של 4 מטרים. בהינתן נתונים אלו, מהו הניחוש הטוב ביותר של מיקום הרובוט? ומהי השונות החדשה?

פתרון:

$$\sigma_3^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{1}{\frac{1}{5} + \frac{1}{4}} = 2.2$$

$$\mu_3 = \frac{\mu_1 \cdot \sigma_2^2 + \mu_2 \cdot \sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{7 \cdot 4 + 11 \cdot 5}{4 + 5} = 9.2$$

תרגיל

על בסיס המיקום האפרורי הראשון של הרובוט (7,5) - פי כמה עלה דיוק המיקום הנוכחי של הרובוט מאשר לפני התזוזה? ניתן להניח לצורך שאלה זו שהדיוק הינו פונקציה ישרה של סטיית התקן.

פתרון:

$$\frac{\sigma_1^2}{\sigma_3^2} = \frac{5}{2.2} = 2.27$$

תרגיל

כשאנחנו מכפלים 2 התפלגויות גאוסיות שהן אותו דבר מה יקרה?
פתרון: הממוצע בריבוע יהיה שווה לחצי מהשונות.

הסתברות לעומת סבירות (Likelihood)

נסביר לפי הביטוי: $\Pr(data | distribution) \neq L(distribution | data)$.

- **הסתברות** - $\Pr$ היא שאלה על ה data בהינתן שההתפלגות ידועה. על מנת לייצר את ההתפלגות אנחנו מנתחים המון data וכאשר מגיע data חדש אנחנו שואלים על ההסתברות שלו ביחס ל data הישן. וכאן נחשב באמת בעזרת אינטגרל. אנחנו יוצאים מנקודה הנחה שיש לנו התפלגות ואז שואלים בהינתן ההתפלגות הזאת מה ההסתברות ל data?
- **סבירות** - L יש לנו כבר data. כאן יש לנו data ואנחנו שואלים מה הסבירות (likelihood) שההתפלגות נכונה ונחפש את הערך $y = f(x)$.

תרגיל

בהינתן ש-IQ מתפלג נורמלית באוכלוסייה עם ממוצע 100 וסטיית תקן של 15, מה הסיכוי (Probability) שאדם אקראי אשר יידגם, יהיה בעל IQ הגבוה מ-125?

פתרון:

$$\Pr(X > 125) = \int_{125}^{\infty} \frac{1}{\sqrt{2\pi \cdot 15^2}} \cdot e^{\frac{-(x-100)^2}{2 \cdot 15^2}} = 0.477$$

תרגיל

נניח שמצאנו אדם בעל IQ 125. מהי הסבירות (Likelihood) שהוא הגיע מתוך אותה התפלגות נורמלית שתיארנו (ממוצע 100, סטיית תקן 15)?

פתרון:

$$L(125) = f(125) = \frac{1}{\sqrt{2\pi \cdot 15^2}} \cdot e^{\frac{-(125-100)^2}{2 \cdot 15^2}} = 0.00663$$

תרגיל

מה הסבירות שהוא הגיע מהתפלגות שונה? (ממוצע 115, סטיית תקן של 20)?

פתרון:

$$L(125) = f(125) = \frac{1}{\sqrt{2\pi \cdot 20^2}} \cdot e^{\frac{-(125-115)^2}{2 \cdot 20^2}} = 0.0176$$

תזוזה אי ודאית (קונבולוציה)

אנו ראינו כבר שתנועה מוסיפה חוסר וודאות למערכת - אך מה קורה בתנועה במסנן קלמן?

- אחד המאפיינים של ניווט עיוור הוא שכלל שהזמן עובר, הדיוק של המערכת הולך ויורד.

התהליך המתמטי אשר מתאר את התופעה הזו נקראת **קונבולוציה**.

נניח ויש לנו גאוס עם σ_1^2 , μ_1 כלשהם המציינים את המיקום של הרובוט שלנו.

כעת נרצה להבין את ההתפלגות של מיקום הרובוט לאחר תנועה.

צריך להבין שהתנועה בעצמה מכילה אי-ודאות כלשהי ולכן היא מתפלגת גאוסית עם σ_2^2 , μ_2 .

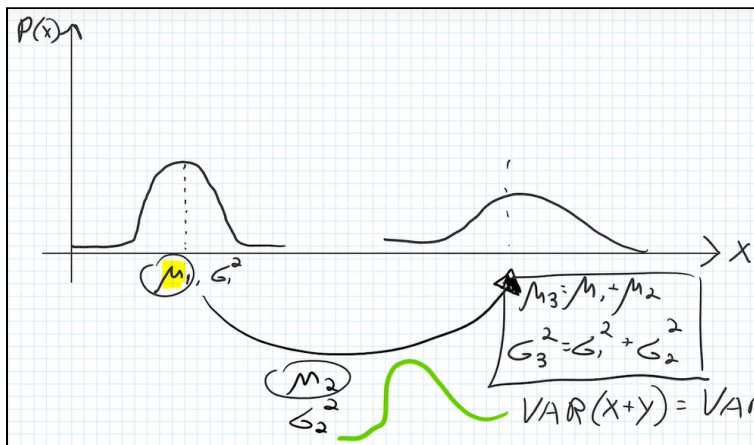
המיקום החדש של הרובוט שלנו הוא:

$$\mu_3 = \mu_1 + \mu_2$$

$$\sigma_3^2 = \sigma_1^2 + \sigma_2^2$$

וזה מתקיים בגלל לינאריות התוחלת ולינאריות השונות:

$$\text{var}(x + y) = \text{var}(x) + \text{var}(y)$$



תרגיל

מיקום רובוט מתאפיין בהתפלגות נורמלית סביב $X=5$ מטרים עם סטיית תקן של 2 מטרים. הרובוט זז 8 מטרים ימינה. לתזוזה יש רעש גאוסית הניתן למידול כסטיית תקן של 1.7 מטרים. מה המיקום החדש של הרובוט? ומה סטיית התקן החדשה שלו?

פתרון:

$$\mu_3 = \mu_1 + \mu_2 = 5 + 8 = 13$$

$$\sigma_3^2 = \sigma_1^2 + \sigma_2^2 = \sqrt{1.7^2 + 2^2} = 2.62$$

תרגיל

לפי חוק ה-95-99-68, מה יהיה הטווח של המיקום החדש ב-95 אחוז מהמקרים?

פתרון:

$$\mu_3 - 2\sigma_3^2 = 13 - 2 \cdot 2.62 = 7.76$$

$$\mu_3 + 2\sigma_3^2 = 13 + 2 \cdot 2.62 = 18.24$$

המודל הדינמי של מסנן קלמן

אני רוצה לדעת גם מה המהירות של הרכב וגם איפה הוא נמצא, העובדה שאני יודע משהו על נתון אחד משפיעה על הנתון השני, זה מודל דינמי של מערכת. המערכת בזמן t שונה מאיפה הייתה בזמן $t-1$, היא זזה, וכדי להבין את זה אני צריך להבין את הפיזיקה או המכניקה או את מה שזה לא יהיה.

ככל שהמודל הדינמי של המערכת שלי מדויק יותר, ככה מסנן קלמן יעבוד טוב יותר.

- במודל סטטי, אי שינוי נרשום כך:

$$x_t = x_{t-1}$$

- במודל דינמי, נשתמש במשוואות התנועה:

$$s = v \cdot t$$

כאשר v זה המהירות ו- t זה הזמן.

נשכלל:

$$x_t = x_{t-1} + \Delta t \cdot v_{t-1}$$

תרגיל

נניח שאנחנו רוצים להעריך גובה של בניין בעזרת מד גובה לא מדויק. מה שאנחנו יודעים בוודאות הוא שגובה הבניין אינו משתנה עם הזמן (לפחות לא במשך המדידה...). בדוגמה שלנו:

- הגובה המדויק של הבניין הוא 50 מטרים.
 - שגיאת המדידה של מד הגובה היא 5 מטרים (סטיית תקן).
- רצף של 10 מדידות עוקבות נתן את הערכים האלה:

מס מדידה	1	2	3	4	5	6	7	8	9	10
תוצאה (מטרים)	48.54	47.11	55.01	55.15	49.89	40.85	46.72	50.05	51.27	49.95

כפי שראינו בפרק, אנחנו מתחילים תמיד מניחוש ראשוני של המצב (STATE) ושל חוסר הוודאות.

במקרה שלנו, ההערכה בעין היא שגובה הבניין הוא 60 מטרים. $\hat{x}_{0,0} = 60m$

נניח שסטיית התקן בהערכת בני אדם היא 15 מטרים. לכן, השונות היא 225 מטר בריבוע. $\sigma^2 = 225$.

בכתיב משוואות קלמן: $p_{0,0} = 225m^2$

מה יהיה קבוע קלמן ושערוך הגובה החדש לאחר המדידה הראשונה (48.54 מטרים)?

פתרון:

$$kg = \frac{15}{15+5} = \frac{15}{20} = \frac{3}{4}$$

$$Est = 60 + \frac{3}{4}(48.54 - 60) = 51.405$$

$$E_{est_t} = (1 - \frac{3}{4}) \cdot (15) = 3.75$$

תרגיל

מה יהיה חוסר הוודאות החדש?

פתרון:

$$kg = \frac{3.75}{3.75+5} = \frac{3}{7} = 0.428$$

$$Est = 51.405 + \frac{3}{7} (47.11 - 51.405) = 49.564$$

$$E_{est_t} = (1 - \frac{3}{7}) \cdot (3.75) = 2.142$$

תרגיל

חזרו על 2 התרגילים הקודמים לגבי המדידה השלישית.

פתרון:

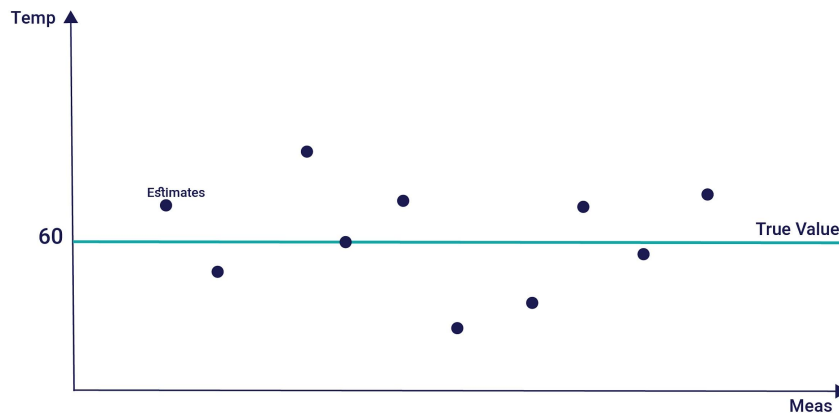
$$kg = \frac{2.142}{2.142+5} = \frac{3}{10} = 0.3$$

$$Est = 49.564 + \frac{3}{10} (55.01 - 49.564) = 51.1978$$

$$E_{est_t} = (1 - \frac{3}{10}) \cdot (2.142) = 1.499$$

מסנן אלפא

ביחידה הקודמת עסקנו בדוגמת שערך הטמפרטורה. כלומר בהינתן סט מדידות מחיישן לא מדויק, האם אנחנו יכולים להעריך בדיוק גבוה את הטמפרטורה האמיתית, וזאת אף על פי שאין לנו דרך לדעת ישירות מהי אלא דרך החיישן? גרף טיפוסי של מדידות יכול להיראות כך:



דרך אחת היא להשתמש בממוצע.

כאשר אין הטיה מסוימת במדידה אלא רק רעש (ממוצע הרעש הוא אפס), אפשר לחשב את הטמפרטורה כך:

$$\hat{x}_{N,N} = \frac{1}{N} (Y_1 + Y_2 + \dots + Y_{N-1} + Y_N) = \frac{1}{N} \sum_{n=1}^N (Y_N)$$

אף על פי שהטכניקה הזאת עובדת, היא בעייתית מכיוון שהיא מחייבת אותנו "להחזיק בזיכרון" את כל המדידות שנעשו. במקום זאת ניתן לבצע את הטריק המתמטי שלהלן ולהגיע לתוצאה הזאת:

$$\boxed{\text{The estimate of current state}} = \boxed{\text{Predicated value of current state}} + \boxed{\text{Factor}} * \left(\boxed{\text{Measurement}} - \boxed{\text{Predicated value of current state}} \right)$$

המשמעות היא שהשערוך בזמן N תלוי רק במדידת ה-N ובשערוך בזמן N-1.

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha_n (y_n - \hat{x}_{n,n-1})$$

בדוגמה שלנו - אלפא הוא המקדם הזה.

סיכום

- בפרק האחרון ראינו את אחד המסננים החשובים ביותר בתורת הבקרה – מסנן קלמן. אמנם ראינו אותו רק בממד אחד (בפרק הבא נראה אותו בממדים נוספים), ובכל זאת למדנו דברים חשובים.
- הדבר הראשון שלמדנו הוא **שקבוע קלמן** (Kalman Gain) מהווה לנו סוג של ממוצע משוקלל בין שני דברים: חיזוי ממצב קודם ומדידה נוכחית מהחיישנים.
 - ככל שהמדידה הנוכחית טובה יותר ומדויקת יותר, כך Kalman Gain יהיה קרוב יותר ל-1.
 - ככל שהחיזוי והמצב הנוכחי (State) מדויק יותר, כך Kalman Gain יהיה קרוב יותר ל-0.
- הרעיון המרכזי הוא ש-Kalman Gain יהיה כמה שיותר קטן, ואז, כאשר תהיה מדידה חדשה, גם אם היא תהיה רועשת, נוכל להכיל אותה בתוך ה-State של המערכת הקיימת.
- דבר נוסף וחשוב מאוד: **מסנן קלמן אינו מחייב אותנו לשמור בזיכרון את כל המדידות שעשינו עד כה**. למעשה, הוא מאפשר לבחון את המדידה האחרונה ואת השערוך האחרון, ועל פי זה **לגזור** את כל המידע.
- נושא נוסף שלמדנו עליו בפרק הקשור למסנן קלמן הוא **ההכפלה**. מסנן קלמן מסתמך על מערכת גאוסית. כלומר כל המדידות צריכות להיות גאוסיות. בהינתן ניחוש המתאפיין בהתפלגות גאוסית, כשנכפיל אותו עם מדידה חדשה שגם לה יש שגיאה גאוסית, נקבל מדידה מדויקת יותר.

מסנן קלמן בשני מימדים

גאוס בשני מימדים

התפלגות רב-נורמלית או התפלגות גאוסיאנית רב-ממדית, היא הכללה של התפלגות נורמלית למשתנים מקריים רב-ממדיים. היא מוגדרת בתור וקטור משתנים מקריים, שכל צירוף ליניארי שלו מתפלג נורמלית.

אם אני מודד רק משתנה אחד, לצורך העניין את X , ואני יודע את הקשר שבין X ל- $\bar{X}$, אני יכול לשערך מהו $\bar{X}$. זה הדבר המשמעותי של **הקורלציה**.

הדבר הזה בא לידי ביטוי בזה שלפעמים החישובים שלי לא מצליחים לבדוק את כל מרחב המצב. אבל מכיוון שיש קשר בין המשתנים במרחב המצב, הקשר הזה יכול לעזור לי כדי לשערך משתנה שאני לא יכול לבדוק.

זו המשמעות הגדולה של התפלגות גאוסית רב-ממדית כאשר יש קורלציה בין המשתנים. הקורלציה הזאת תבוא לידי ביטוי במודל הדינמי של המערכת או **במטריצת F** .

קורלציה בין משתנים

- חישוב השונות:

$$\sigma_x^2 = \frac{\sum_{i=1}^N (\bar{x} - x_i)^2}{N}$$

כאשר הממוצע הוא $\bar{x}$.

- סטיין התקן:

$$\sigma_x = \sqrt{\sigma_x^2}$$

כלומר סטיית התקן היא השורש של השונות.

- שונות משותפת:

$$\sigma_x \sigma_y = \frac{\sum_{i=1}^N (\bar{x} - x_i)(\bar{y} - y_i)}{N}$$

משוואות התנועה

משוואות התנועה הן בעצם הדרך המתמטית לתאר אובייקט שזז שזה יכול להיות כל אובייקט וזה גם יכול להיות רכב.

- מהירות קבועה:

$$V(t) = V_0$$

$$X(t) = V_0 \cdot t + X_0$$

- תאוצה קבועה:

$$a(t) = a_0$$

$$V(t) = a_0 \cdot t + V_0$$

$$X(t) = V_0 \cdot t + \frac{a_0 \cdot t^2}{2} + X_0$$

כאשר $a(t)$ התאוצה הקבועה. מכיוון שהתאוצה היא השינוי של המהירות, אז האינטגרל על התאוצה אמור לתת את המהירות. ה- V_0 זאת המהירות ההתחלתית לפני שהתחלנו בתאוצה. המשוואות שראינו עכשיו, מתארות תנועה במימד אחד אבל ניתן להכליל אותן למימדים נוספים. כל המשוואות האלו ינסו לטפל ב- Δt , כלומר בשינוי שקרה מהמצב האחרון למצב הנוכחי שאותו אנחנו רוצים לשערך.

דוגמאות:

1. רכב נוסע במהירות קבועה של 5 מטר בשנייה. נניח שהרכב התחיל את הנסיעה שלו ב $X = 14$, מה יהיה המיקום שלו לאחר 11 שניות?

פתרון:

נציב:

$$X(11) = 5 \cdot 11 + 14 = 69$$

2. רכב עומד בנקודה $X = 0$ ומאיץ בתאוצה של 4.5 מטר בשנייה בריבוע. לאחר כמה זמן יגיע הרכב לנקודה $X = 100$?

פתרון:

נציב:

$$100 = \frac{1}{2} \cdot 4.5 \cdot t^2$$

$$t = 6.66$$

שלב ה-Prediction

שלב ה-Prediction זה בעצם השלב שלוקח את המצב הנוכחי שלנו ומנסה לשערך מה יהיה בעתיד על בסיס משוואות התנועה והמודל הדינמי של המערכת.

חלק ראשון

לגאוס במימד אחד נזדקק ל- μ , σ^2 .
לגאוס בכמה מימדים נזדקק ל- μ , Σ , כאשר:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots \\ \sigma_{2,1} & \sigma_2^2 & \\ \vdots & & \ddots & \\ & & & \sigma_n^2 \end{pmatrix}$$

נוכל להגדיר μ כ- **וקטור המצב** - הערכים שאותם אנחנו רוצים לבדוק ולשערך למשל נסמן $\mu = \vec{x}_k$ כאשר:

$$\vec{x}_k = \begin{pmatrix} \text{Position} \\ \text{Velocity} \end{pmatrix}$$

נגדיר את $\Sigma = P$ כלומר P היא מטריצת ה-Covariance:

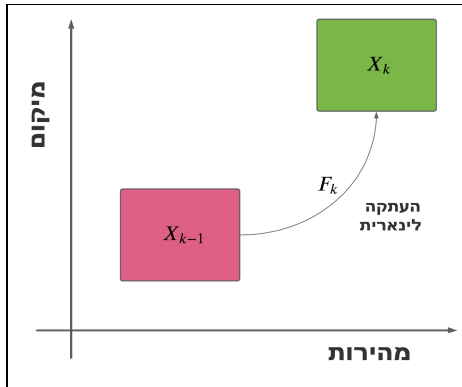
$$P = \begin{pmatrix} \sigma_{pos}^2 & \sigma_{pos}\sigma_{vel} \\ \sigma_{vel}\sigma_{pos} & \sigma_{vel}^2 \end{pmatrix}$$

כעת נרצה להשתמש בכלים אלו (הגאוסין המטריציוני + משוואות התנועה על מנת לתאר תנועה במהירות קבועה), כלומר תרגמנו את התזוזה למשוואת התנועה וכעת נכיל את הכתיב המטריציוני:

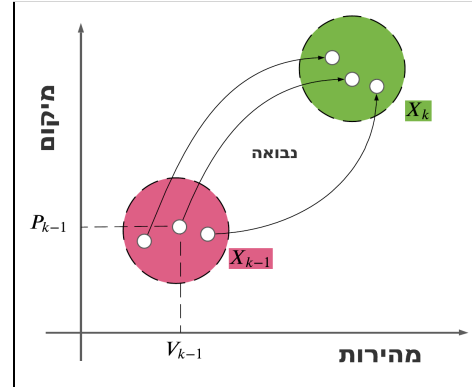
$$\begin{matrix} P(k+1) = P(k) + \Delta t \cdot v_k \\ V(k+1) = V(k) \end{matrix} \iff \begin{pmatrix} P \\ V \end{pmatrix}_{k+1} = \underbrace{\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}}_{F \text{ (in Kalman)}} \begin{pmatrix} P \\ V \end{pmatrix}_k$$

במטריצה F קיים Δt כיוון שהמהירות משפיעה על מרחב המצב שלנו.

ולמעשה מסנן קלמן יכול לקחת מודלים לינאריים ולתרגם למשוואות לינאריות, כלומר מטריצות ולבצע את המעבר לצעד הבא על ידי התמרה לינארית F_k (העתקה לינארית) כך:



נסתכל על השלב הזה בצורה גרפית: בגרף כל נקודה מתארת מהירות ומיקום. כל נקודה בורוד מייצגת את השערוך שלי למיקום/מהירות ה- $k - 1$ וכעת נרצה "להעביר" כל נקודה לצעד ה- k .



נעזר בכללים הבאים:

$$\begin{aligned} cov(x) &= \Sigma \\ cov(A \cdot x) &= A \cdot \Sigma \cdot A^T \end{aligned}$$

וכיוון ש- P היא מטריצת ה-Covariance (האי וודאות) אז מתקיים:

$$\begin{aligned} X_k &= F \cdot X_{k-1} \\ P_k &= F \cdot P_{k-1} \cdot F^T \end{aligned}$$

- המשוואה הראשונה מעבירה אותנו מהמצב הנוכחי $k - 1$ אל המצב הבא k . כלומר מווקטור המצב הנוכחי X_{k-1} אל וקטור המצב הבא X_k בעזרת טרנספורמציה לינארית עם מטריצת F .
- המשוואה השנייה מתארת לנו את חוסר הוודאות החדש, אנחנו נרצה לא רק לדעת איפה אנחנו נמצאים אלא גם מה חוסר הוודאות החדש שלנו.

חלק שני

נרצה להתמודד עם שני אלמנטים חשובים:

1. שינויים ידועים.
2. רעש סטוכסטי (אקראי).

שינויים ידועים:

וקטור המצב מתאר לנו את כל הפרמטרים שאנחנו רוצים לדעת עליהם משהו. אבל לפעמים יש שינויים בתוך המערכת שלא קשורים לווקטור המצב למשל תאוצה/תאוצה/סיבוב ההגה. כל שינוי שאנחנו יודעים עליו יכול להיכנס לוקטור הבקרה $\vec{u}_k$ ובכך לשפר את הדיוק שלנו:

דוגמה:

נניח שאנחנו יודעים שיש לנו תאוצה a_0 . משוואות התנועה הם:

$$P_k = P_{k-1} + \Delta t \cdot V_{k-1} + \frac{1}{2} \cdot a_0 \cdot \Delta t^2$$

$$V_k = V_{k-1} + \Delta t \cdot a_0$$

משוואות התנועה אלה בעצם הגורמים שאנו מוסיפים כאשר יש לנו במקום מהירות קבועה - תאוצה קבועה. היתרון במשוואות אלה הם שאפשר לכתוב אותם בצורה לינארית. את המשוואות האלה אפשר לכתוב בכתיב מטריציוני כזה:

$$\begin{pmatrix} P \\ V \end{pmatrix}_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} P \\ V \end{pmatrix}_{k-1} + \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix} \cdot a_0$$

בכתיב של קלמן המטריצה שהצגנו למעלה $\begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix}$ נסמן B ונקראת **מטריצת הבקרה**.

הוקטור a_0 נסמן $\vec{u}$ ונקרא לו **וקטור הבקרה**. ומהדבר הזה אנחנו מגיעים למשוואת Prediction שהיא יותר מדויקת ונכונה ומכילה יותר אלמנטים וכך נוכל לעשות שיערוך (אסטמציה) יותר טוב לוקטור המצב בזמן הבא והמשוואה תראה כך:

$$X_k = F \cdot X_{k-1} + B \cdot \vec{u}$$

כלומר, המשוואה מתארת טרנספורמציה לינארית מ- X_{k-1} ל- X_k ובנוסף, אנחנו מתחשבים בשינויים שאנחנו מכירים כמו תאוצה או תאוצה ופרמטרים אחרים. הסיבה שאין במשוואה הזאת רעש היא מכיוון שבכל מסנן קלמן אנחנו מניחים שהרעש מתפלג סביב נקודה 0. וכאשר אנחנו נחשב את הממוצע, שזה בעצם ווקטור המצב שלנו, אין לנו רעש בכלל (הוא פשוט אפס).

רעש סטוכסטי (אקראי):

- נדבר על שינויים שאנחנו לא יודעים מה הם, אבל הם משפיעים לנו על המיקום של הרכב.
- בכל פעם שמטריצת F מעבירה אותנו מהמצב הנוכחי אל המצב הבא, יש חוסר וודאות שצריך לתת לה ביטוי מתמטי.
- בדוגמה ריאליסטית, כאשר השטח שלנו הוא חולי והרכב יכול להחליק, אנחנו נצטרך להכניס את המציאות הזאת לתוך המתמטיקה ולתאר את האזור האמורפי (מקרה כאשר מטריצה F מעבירה אותנו לאזור עם חוסר וודאות) שהנקודה מגיעה אליו. הדרך המתמטית לתאר את זה היא דרך מטריצה שנקרא לה Q .
- מטריצת Q היא בעצם מטריצה שאומרת לנו שיש לנו חוסר וודאות שהוא נוסף לחוסר הוודאות שהיה כבר בתוך המערכת כפי שבא לידי ביטוי במטריצת P הקודמת.
- מטריצה Q מוסיפה חוסר וודאות בתזוזה.

על פי משוואות התנועה יצרנו את המודל:

$$\begin{pmatrix} P'_x \\ P'_y \\ V'_x \\ V'_y \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_F \begin{pmatrix} P_x \\ P_y \\ V_x \\ V_y \end{pmatrix} + \underbrace{\begin{pmatrix} a_x \frac{\Delta t^2}{2} \\ a_y \frac{\Delta t^2}{2} \\ a_x \Delta t \\ a_y \Delta t \end{pmatrix}}_U$$

מה נעשה כאשר אין לנו תאוצה ונצטרך לשערך את התאוצה?

- נסתכל על $N(0, Q) \sim \mathbf{u}$ כעל "רעש סטוכסטי / אקראי", נסמן את מטריצת ה-Covariance ב- Q . כיוון שמדובר ברעש נוכל להסתכל עליו בהתפלגות נורמלית עם ממוצע של אפס.
- הרעש הסטוכסטי, כלומר הוקטור $\mathbf{u}$ מכיל את השונות ואת התאוצות a_x, a_y כאשר x, y הם הכיוונים בתאוצה. את וקטור התאוצה (הצד השמאלי בשוויון) ניתן לכתוב כמכפלה של שתי מטריצות אחת בשנייה.
- המטריצה G הדטרמיניסטית, כלומר שכל האיברים שנמצאים בה הם איברים שאנחנו יודעים מה הם. המטריצה a היא המטריצה האקראית, אנחנו לא מכירים את a_x, a_y מכיוון שזה תאוצה של אובייקט חיצוני כלשהי שאין לנו מידע לגביו.

$$\begin{pmatrix} a_x \frac{\Delta t^2}{2} \\ a_y \frac{\Delta t^2}{2} \\ a_x \Delta t \\ a_y \Delta t \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix}}_{\text{Deterministic}(G)} \underbrace{\begin{pmatrix} a_x \\ a_y \end{pmatrix}}_{\text{Random}(a)} = Ga$$

כעת צריך לחשב את המטריצה Q כך:

$$Q = E[(u - E(u)) \cdot (u - E(u))^T]$$

מכיוון ש- u מתפלג עם ממוצע של 0, אז התוחלות של u מתאפסים:

$$Q = [u \cdot u^T] = E[Ga \cdot (Ga)^T]$$

כעת, מאחר ש- G דטרמיניסטית אז היא יוצאת מהתוחלת (כי היא ידועה, כלומר ערך קבוע ידוע) ונשאר לנו:

$$= G \cdot E[aa^T] \cdot G^T$$

התוחלת הנתונה $E[aa^T]$ היא בעצם מטריצת ה- Cov של a , כלומר:

$$= G \cdot \begin{pmatrix} \sigma_{ax}^2 & \sigma_{axy} \\ \sigma_{ayx} & \sigma_{ay}^2 \end{pmatrix} \cdot G^T$$

נשים לב שאין קשר בין התאוצה של האובייקט החיצוני שזז בציר x כלומר σ_{ax}^2 לבין האובייקט החיצוני שזז בציר

y כלומר σ_{ay}^2 ולכן, $\sigma_{axy} = 0$, $\sigma_{ayx} = 0$ כי אין ביניהם במשותף גם ככה. לכן:

$$= G \cdot \begin{pmatrix} \sigma_{ax}^2 & 0 \\ 0 & \sigma_{ay}^2 \end{pmatrix} \cdot G^T$$

כלומר נקבל: $Q = G \cdot Q_u \cdot G^T$ ואת המטריצה הזאת אנחנו נוסיף לווקטור P כאשר נחשב את הרעש.

כשנאסוף את כל המידע נגיע לשתי משוואות שסוגרות לנו את כל שלב השערוך או ה-Prediction במסגרת קלמן.

$$X_k = F \cdot X_{k-1} + B \cdot \vec{u}$$

$$P_k = F_k \cdot P_{k-1} \cdot F_k^T + Q_k$$

- משוואה ראשונה: השערוך הטוב ביותר שלנו עבור X_k זה השערוך שנוצר כתוצאה מהמצב הקודם

X_{k-1} כולל התיקון של השפעות ידועות שחיצוניות למערכת $B \cdot \vec{u}$ (כמו תאוצה למשל).

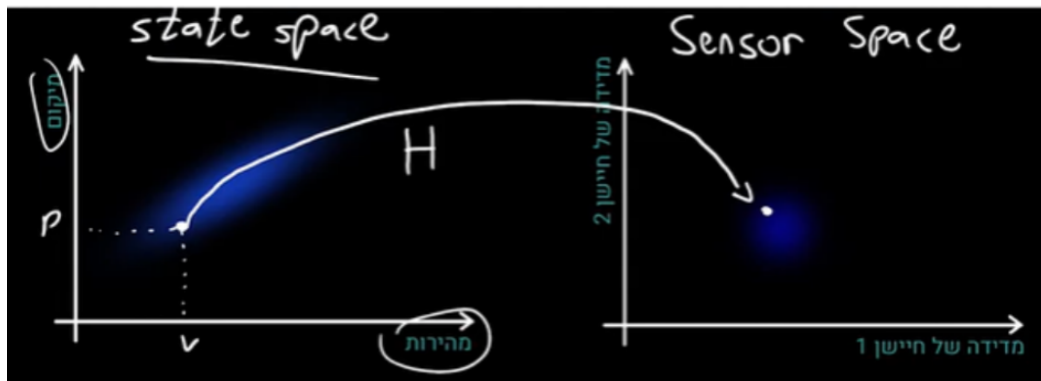
- משוואה שנייה: ה- P_k מתאר את חוסר הוודאות במערכת שלנו $F_k \cdot P_{k-1} \cdot F_k^T$ יחד עם חוסר וודאות

מהסביבה החיצונית שזה אותו ה- Q_k .

שלב ה-Measurement

חלק ראשון:

בעיה - לא תמיד החיישן מכיל את המידע שאנו צריכים ל- X_k .



הגרף השמאלי מתאר $state - space$, הימני $sensor - space$ ואנחנו נרצה ליצור טרנספורמציה בין המצב הקיים לבין המידע שאני מקבל, נעשה זאת על ידי מטריצה H .

דוגמה:

- נניח ויש לנו וקטור עם מיקום ומהירות וחיישן של מיקום אז:

$$H \cdot X = \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \text{Position} \\ \text{Velocity} \end{pmatrix} = (\text{Position})$$

- נשים לב שאם יש לנו חיישן שבודק גם מיקום וגם מהירות אז:

$$H = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- אם $H = \begin{pmatrix} 1 & 1 \end{pmatrix}$ אז:

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \text{Position} \\ \text{Velocity} \end{pmatrix} = (\text{Position} + \text{Velocity})$$

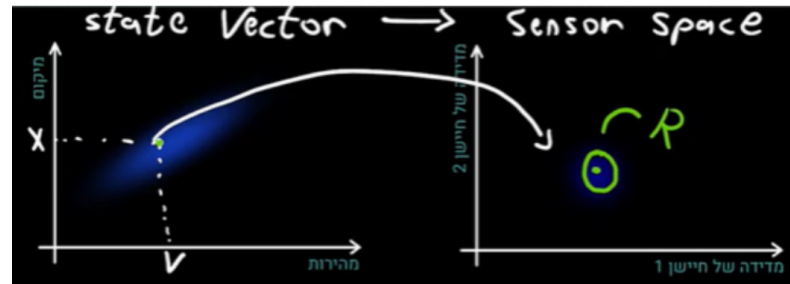
כלומר, החיישן יודע לחשב מיקום + מהירות.

- לסיכום:

$$\mu_{Expected} = H_k \cdot X_k$$

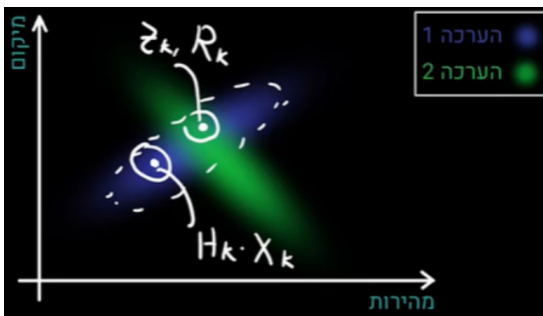
$$\Sigma_{Expected} = H_k \cdot P \cdot H_k^T$$

שתי המשוואות האלה בעצם לוקחות את שלב ה-Prediction ומעבירות אותו למרחב החיישן.

חלק שני:

המעבר ממרחב המצב (State Space) למרחב החיישן (Sensor Space) מתבצעת על ידי $H \cdot X$ והיא נותנת לנו מטריצה שנסמן אותה ב- R .

המטריצה R היא מטריצת הרעש שנוצר מהמעבר בין מרחב המצב לבין מרחב החיישן.



נרצה לדעת לסנכרן בין המדידה להערכה (וקטור המצב).
בציור נניח שהכחול היא ההערכה והירוק היא המדידה.
את הסנכרון הזה נעשה על ידי מסנן קלמן, רק שעכשיו
נצטרך לעלות במימדים.

נעשה סידור במשוואות שהכרנו כבר ונקבל את השלב הבא שנקרא **שערוך אחרי מדידה:**

$$\begin{aligned} X'_k &= X_k + K' \cdot (\vec{Z}_k - H_k \cdot X_k) \\ P'_k &= P_k - K' \cdot H_k \cdot P_k \\ K' &= P_k \cdot H_k^T \cdot (H_k \cdot P_k \cdot H_k^T + R_k)^{-1} \end{aligned}$$

מבט על:

1. בשלב הראשון היה לנו Prediction - כלומר אני יודע איפה אני נמצא עכשיו, אני מכיר את המודל הדינמי והעברתי אותו קדימה.
 2. בשלב השני היה לנו מדידה Measurement.
 3. בשלב השלישי היה לקחת את המדידה ואת הפרדיקציה, ונחבר אותם ביחד לפי הכפלת גאוסיאנים ולקבל את השערוך החדש.
- ברגע שיש לנו את חוסר הוודאות של הפרדיקציה והמדידה אז אנחנו יכולים להסיק מזה בדיוק טוב את השאלה - היכן הרכב שלנו באמת נמצא (לפי הכפלת גאוסיאנים).
 - מסנן קלמן הוא מסנן רקורסיבי.
 - במשוואה הראשונה X'_k זה השערוך החדש - מה שאנחנו משערכים.

סיכום

- בפרק הזה ראינו איך עובד מסנן קלמן בכמה ממדים והבנו את הכתיבה המטריציונית.
- כמו שהסברנו, כל המשוואות המסובכות הן בעצם הכללה לממדים נוספים של המשוואות הפשוטות שראינו בפרק הקודם.
- שלב ה-Prediction זה בעצם השלב שלוקח את המצב הנוכחי שלנו ומנסה לשערך מה יהיה בעתיד על בסיס משוואות התנועה והמודל הדינמי של המערכת.

$$X_k = F \cdot X_{k-1}$$

$$P_k = F \cdot P_{k-1} \cdot F^T$$

שינויים ידועים:

נסמן B מטריצת הבקרה.

נסמן $\vec{u}$ ונקרא לו וקטור הבקרה.

ומהדבר הזה אנחנו מגיעים למשוואת Prediction טובה:

$$X_k = F \cdot X_{k-1} + B \cdot \vec{u}$$

רעש סטוכסטי (אקראי):

$$X_k = F \cdot X_{k-1} + B \cdot \vec{u}$$

$$P_k = F_k \cdot P_{k-1} \cdot F_k^T + Q_k$$

- שלב ה-Measurement

שתי המשוואות האלה בעצם לוקחות את שלב ה-Prediction ומעבירות אותו למרחב החיפוש.

$$\mu_{Expected} = H_k \cdot X_k$$

$$\Sigma_{Expected} = H_k \cdot P \cdot H_k^T$$

- שערך אחרי מדידה:

$$X_k' = X_k + K' \cdot (\vec{Z}_k - H_k \cdot X_k)$$

$$P_k' = P_k - K' \cdot H_k \cdot P_k$$

$$K' = P_k \cdot H_k^T \cdot (H_k \cdot P_k \cdot H_k^T + R_k)^{-1}$$

- בפרק הבא נתמודד עם אחת הבעיות הקשות של מסנן קלמן – הדרישה ללינאריות.
- נלמד איך ניתן להשתמש במסנן קלמן גם כאשר פונקציית המעבר (או החיפוש) אינה לינארית.

אלגוריתם קלמן

1. אתחול - נציב נתונים.

$$\begin{pmatrix} X_0 \\ P_0 \end{pmatrix}$$

2. חישוב משוואות ה-Prediction -

$$\begin{pmatrix} X_k = A \cdot X_{k-1} + B \cdot \vec{u}_k + w_r \\ P_{k_p} = A \cdot P_{k-1} \cdot A^T + Q_k \end{pmatrix}$$

a. חישוב מצב ה-Prediction של משוואה X_k b. אתחול מטריצת ה-Cov - נעשה זאת רק פעם אחת (P_{k-1}).c. חישוב משוואה P_{k_p} - עדכון + רעש חיצוני (Q_k).3. התאמת מצב ה-Prediction -

$$\begin{pmatrix} K = \frac{P_k \cdot H^T}{H \cdot P_k \cdot H^T + R} \\ Y_k = X_{k_p} + K \cdot \begin{bmatrix} Z_k - H \cdot X_{k_p} \end{bmatrix} \end{pmatrix}$$

a. חישוב הגבר קלמן - כולל מעבר ממרחב המצב למרחב החיפוש H עם תוספת רעש R .b. קבלת מדידה חדשה: מעבר ממרחב החיפוש למרחב המצב $C + z$ לפי המשוואה:

$$\begin{pmatrix} Y_k = C \cdot Y_{k_m} + z_m \end{pmatrix}$$

c. חישוב מצב נוכחי - התאמת X_k .4. עדכון מטריצת ה-Cov -

$$\begin{pmatrix} P_k = (I - K \cdot H) \cdot P_k \\ X_k \end{pmatrix}$$

5. הכנה לסיבוב הבא - מעדכנים את הוקטור משלב (1) לפי הנתונים החדשים שהתקבלו בסיבוב הזה.6. חוזרים לשלב (2).

תרגיל סיכום אלגוריתם קלמן

וקטור המצב הוא מיקום ומהירות בציר X . ניתן להניח שדלתא T שווה 1. הסתכלו על הנתונים שלפניכם וענו על השאלות מטה:

Given:

$$V_{0X} = 280 \text{ m/sec } X_0 = 4000\text{m}$$

$$V_{0Y} = 120 \text{ m/sec } Y_0 = 3000\text{m}$$

Observations:

$X_0 = 4000\text{m}$	$V_{0X} = 280\text{m/sec}$
$X_1 = 4260\text{m}$	$V_{1X} = 282\text{m/sec}$
$X_2 = 4550\text{m}$	$V_{2X} = 285\text{m/sec}$
$X_3 = 4860\text{m}$	$V_{3X} = 286\text{m/sec}$
$X_4 = 5110\text{m}$	$V_{4X} = 290\text{m/sec}$

Initial Conditions:

$$A_x = 2\text{m/sec}^2$$

$$\Delta t = 1 \text{ sec}$$

$$V_x = 280\text{m/sec}$$

$$\Delta X = 25\text{m}$$

Process Errors in Process Covariance Matrix:

$$\Delta P_x = 20\text{m}$$

$$\Delta P_{vx} = 5\text{m/sec}$$

Observation Errors:

$$\Delta X = 25\text{m}$$

$$\Delta V_x = 6\text{m/sec}$$

פתרון לפי שלבי האלגוריתם

1. אתחול -

$$\begin{pmatrix} X_0 \\ P_0 \end{pmatrix} = \begin{pmatrix} X_0 \\ V_{0,X} \end{pmatrix} = \begin{pmatrix} 4000 \\ 280 \end{pmatrix}$$

2. חישוב משוואות ה-Prediction -

a. חישוב מצב ה-Prediction של משוואה X_k :

$$\begin{aligned} X_1 &= A \cdot X_0 + B \cdot \vec{u}_1 + w_r = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_0 \\ V_{0,X} \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \cdot \Delta t^2 \\ \Delta t \end{pmatrix} \cdot (a_{X_0}) + 0 = \\ &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 4000 \\ 280 \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \cdot 1 \\ 1 \end{pmatrix} \cdot (2) = \begin{pmatrix} 4000+280 \\ 280 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 4281 \\ 282 \end{pmatrix} = X_{1_p} \end{aligned}$$

b. אתחול המטריצה Cov (חד פעמי) - בדוגמה אין השפעה בין המיקום למהירות ולכן נשמיט בסוף החישוב כדי להקל להמשך הפתרון:

$$P_0 = \begin{pmatrix} \Delta x^2 & \Delta x \Delta V_x \\ \Delta V_x \Delta x & \Delta V_x^2 \end{pmatrix} = \begin{pmatrix} 20^2 & 0 \\ 0 & 5^2 \end{pmatrix} = \begin{pmatrix} 400 & 0 \\ 0 & 25 \end{pmatrix}$$

c. חישוב משוואה P_{k_p} - עדכון + רעש חיצוני (Q_k) , נניח שאין רעש לכן $Q = 0$:

$$\begin{aligned} P_{1_p} &= A \cdot P_0 \cdot A^T + Q_1 = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 400 & 0 \\ 0 & 25 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ \Delta t & 1 \end{pmatrix} + 0 = \\ &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 400 & 0 \\ 0 & 25 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 425 & 25 \\ 25 & 25 \end{pmatrix} \Rightarrow \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \end{aligned}$$

3. התאמת מצב ה-Prediction -

a. חישוב הגבר קלמן - כולל מעבר ממרחב המצב למרחב החישה H עם תוספת רעש R :

עבור H , החישה יודע לקבל את המהירות והמיקום ולכן אין צורך לשנות מימד.

עבור R מטריצת הרעש המדידה נתאים לפי הנתון:

$$R = \begin{pmatrix} 25^2 & 0 \\ 0 & 6^2 \end{pmatrix} = \begin{pmatrix} 625 & 0 \\ 0 & 36 \end{pmatrix}$$

נחשב את K :

$$\begin{aligned} K &= \frac{P_{1_p} \cdot H^T}{H \cdot P_{1_p} \cdot H^T + R} = \frac{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 625 & 0 \\ 0 & 36 \end{pmatrix}} = \frac{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix}}{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} + \begin{pmatrix} 625 & 0 \\ 0 & 36 \end{pmatrix}} = \\ &= \frac{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix}}{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} + \begin{pmatrix} 625 & 0 \\ 0 & 36 \end{pmatrix}} = \frac{\begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix}}{\begin{pmatrix} 1050 & 0 \\ 0 & 61 \end{pmatrix}} = \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \cdot \begin{pmatrix} 1050 & 0 \\ 0 & 61 \end{pmatrix}^{-1} = \end{aligned}$$

$$= \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \cdot \left(\frac{1}{1050 \cdot 61 - 0} \cdot \begin{pmatrix} 61 & 0 \\ 0 & 1050 \end{pmatrix} \right) = \begin{pmatrix} 0.405 & 0 \\ 0 & 0.410 \end{pmatrix}$$

b. קבלת מדידה חדשה: מעבר ממרחב החייושן למרחב המצב $C + z$ לפי המשוואה:

תחילה נוודא שבמדידה החדשה יש התאמה למימד הנוכחי שלנו.

$$\text{לפי הנתון: } X_1 = 4260, \quad V_{X_1} = 282$$

לכן נחשב:

$$Y_1 = C \cdot Y_{1_m} + z_m = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4260 \\ 282 \end{pmatrix} = \begin{pmatrix} 4260 \\ 282 \end{pmatrix}$$

c. חישוב מצב נוכחי - התאמת X_k .

$$\begin{aligned} X_1 &= X_{1_p} + K \cdot \left[Y_1 - H \cdot X_{1_p} \right] = \\ &= \begin{pmatrix} 4281 \\ 282 \end{pmatrix} + \begin{pmatrix} 0.405 & 0 \\ 0 & 0.410 \end{pmatrix} \cdot \left[\begin{pmatrix} 4260 \\ 282 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4281 \\ 282 \end{pmatrix} \right] = \\ &= \begin{pmatrix} 4281 \\ 282 \end{pmatrix} + \begin{pmatrix} 0.405 & 0 \\ 0 & 0.410 \end{pmatrix} \cdot \begin{pmatrix} -21 \\ 0 \end{pmatrix} = \begin{pmatrix} 4281 \\ 282 \end{pmatrix} + \begin{pmatrix} -8.5 \\ 0 \end{pmatrix} = \begin{pmatrix} 4272.5 \\ 282 \end{pmatrix} \end{aligned}$$

4. עדכון מטריצת ה-Cov -

$$\begin{aligned} P_k &= (I - K \cdot H) \cdot P_k = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.405 & 0 \\ 0 & 0.410 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} \\ &= \begin{pmatrix} 0.595 & 0 \\ 0 & 0.590 \end{pmatrix} \begin{pmatrix} 425 & 0 \\ 0 & 25 \end{pmatrix} = \begin{pmatrix} 253 & 0 \\ 0 & 14.8 \end{pmatrix} \end{aligned}$$

5. הכנה לסיבוב הבא - מעדכנים את הוקטור משלב (1) לפי הנתונים החדשים שהתקבלו בסיבוב הזה.

6. חוזרים לשלב (2).

מסנן קלמן מורחב

מבוא לפרק

- חיישנים** - הדבר הראשון שנעשה זה, נכיר את החיישנים, נכיר את החיישנים שיש ברכב האוטונומי, ויותר מזה, נתמקד בשני חיישנים שאיתם נעבוד, חיישן **Radar** וחיישן **LiDAR**, ונסתכל על היתרונות והחסרונות של כל אחד מהם.
- לינארזציה** - הדבר השני זה, נסתכל על הצד המתמטי, על הלינארזציה, איך הדברים האלה עובדים. מה זה **מטריצת היעקוביאן**, מה זה **פיתוח לפי טיילור**, ויותר חשוב מזה, איך מתכנתים את כל זה לכדי משהו עובד שנוכל להשתמש בו.
- היתוך חיישנים** - הדבר האחרון שנעשה זה Sensor Fusion, **היתוך חיישנים**. מה קורה כאשר יש לי חיישן אחד שאומר לי משהו מסוים, חיישן שני שאומר לי משהו אחר? איך אני יכול לערבב, או איך אני יכול להתיק את שני החיישנים או את המידע שמגיע משני החיישנים, לידי ניחוש או שערוך שיהיה טוב יותר מכל אחד מהחיישנים בנפרד?

סוגי חיישנים

עבור תנועה לא לינארית, מסנן קלמן לא יביא תוצאה אופטימלית. מה נעשה עם חיישנים שלא יודעים להחזיר תשובה לינארית? כיצד נתמודד? - הפתרון זה שימוש בקלמן מורחב שמשתמש בקירוב של פונקציות.

חיישן GPS

- יתרונות: זול ונפוץ, כינמי.
- חסרונות: שימושי רק בחוץ ולא מספיק מדויק.

חיישן IMU

מכיל:

- מד תאוצה
- מד מהירות זוויתית
- מגנטומטר - משתמשים בשביל לזהות "מנח"

חיישן LiDAR

נסביר בהמשך

חיישן ה-LiDAR

- מזהה בעזרת קרן לייזר את העצמים - מבצע סריקה תלת-מימדית של המרחב.
- זה אותו מכשיר גדול ומסתובב שנמצא על הגג של הרכב האוטונומי.
- השימוש המקצועי ביותר מגניב הראשון היה בתוכנית החלל "אפולו 15", שם השתמשו ב-LiDAR כדי למפות את פני השטח של הירח, איפה יש גבעות, איפה יש בקעות וכן הלאה.

עקרון הפעולה של החיישן

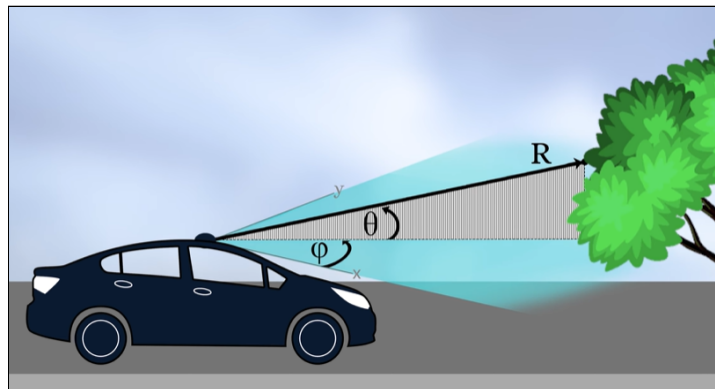
- החיישן עובד בעקרון שנקרא Time Of Flight שזה בעצם חישוב משך הזמן שלוקח לאות להגיע.
 - כדי לעשות LiDAR כל מה שאני צריך זה לייזר, חיישן אור ושעון מאוד מאוד מדויק.
1. ה-LiDAR שולח קרן לייזר עד לאובייקט, הקרן מגיעה, פוגעת ומתפזרת לכל הכיוונים. בין היתר היא חוזרת בכיוון הנגדי של כיוון ההגעה שלה ואז ה-LiDAR קולט אותה בחזרה באמצעות חיישן האור.
 2. השעון המדויק יכול למדוד כמה זמן עבר מהרגע ששלחנו את הקרן עד הרגע שקיבלנו אותה בחזרה.

השיטה לקבלת מרחק מנקודות

אם ידועה לי מהירות האור c , המרחק ביני לבין האובייקט הוא חצי t כפול c מהירות האור: $R = \frac{c \cdot t}{2}$

השיטה הזאת מתקיימת רק לנקודה אחת, מה קורה אם נרצה לקבל מרחק לכל הנקודות במישור? לכן על גג המכונית, איפה שה-LiDAR נמצא, נציב מראה שמסתובבת לכל הכיוונים עם מנוע וככה נוכל לדגום את כל המרחב לכל כיוון: ימינה, שמאלה, למעלה, למטה וכו'... לכן, עבור כל נקודה יש לנו שלושה פרמטרים:

1. המרחק Range.
 2. האזימוט.
 3. האלביציה - Elevation.
- באמצעות שלושת הפרמטרים האלה אנחנו יכולים לעבור לקואורדינטות קרטזיות x, y, z בעצם ייצגו את כל הכיוונים (כל המרחב כולו מסביב לרכב) וכך נוכל לדגום את כל הנקודות.



מעבר לקואורדינטות קרטזיות

המעבר שממיר מקואורדינטות ספריות לקואורדינטות קרטזיות הם בעזרת הנוסחאות הבאות:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right)$$

$$\phi = \cos^{-1}\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) = \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

ענן הנקודות

אנחנו מקבלים הרבה מאוד נקודות במרחב בעזרת ה-LiDAR.

למקבץ הנקודות הזה במרחב אנחנו קוראים **Point Cloud** - שזה מה ש-LiDAR מספק לנו.

מה שניתן לעשות עם ענן הנקודות הוא:

1. לשחזר תמונה תלת מימדית כמעט כמו בתמונה.
2. רגיסטרציה - זה להבין שאותו סט נקודות שתיארו אובייקט שהיה לנו במצב $t = 0$ גם יתאר את אותו האובייקט במצב $t = 1$.

שאלה

מה הם הדברים שצריך לקחת בחשבון כאשר מיישמים חיישן LiDAR עבור רכבים אוטונומיים?

תשובה

1. חיישן ה-LiDAR צריך תמיד להיות ממוקם אופקית, שכן אפילו הטיה קטנה תגרום לשגיאות מדידה.
2. כאשר הרכב נע במהירות, חשוב להביא בחשבון את הפרשי הזמן בין פעימות חיישן ה-LiDAR.
3. חשוב לזהות עצמים מבריקים ובוהקים מאוד בסביבה, אחרת מדידות חיישן ה-LiDAR של אותם אובייקטים עשויות להיות לא תקפות.

חיישן ה-RADAR (מכ"מ)

חיישן הרדאר מאפשר לדעת את המהירות של האובייקט שאנחנו עוקבים אחריו. אנחנו רוצים לשערך איפה האובייקט יהיה ברגע הבא כדי לא לדרוס אותו.

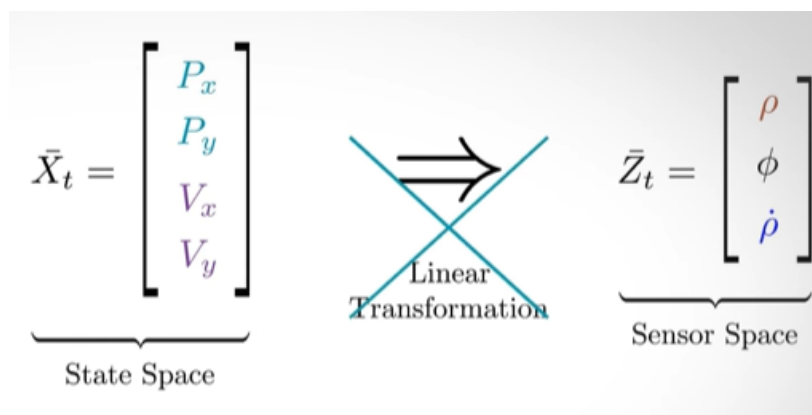
אפקט דופלר

חיישן הרדאר מבוסס על עיקרון פיזיקלי שנקרא "אפקט דופלר". כאשר הגוף יתקרב אלינו, נתפוס את התדר שלו כאילו הוא עולה. כאשר הגוף מתרחק מאיתנו, נתפוס את התדר שלו כאילו הוא יורד. (מאותו סיבה שסירנה של אמבולנס שכלל שהיא מתרחקת מאיתנו הטונים שלה יורדים).

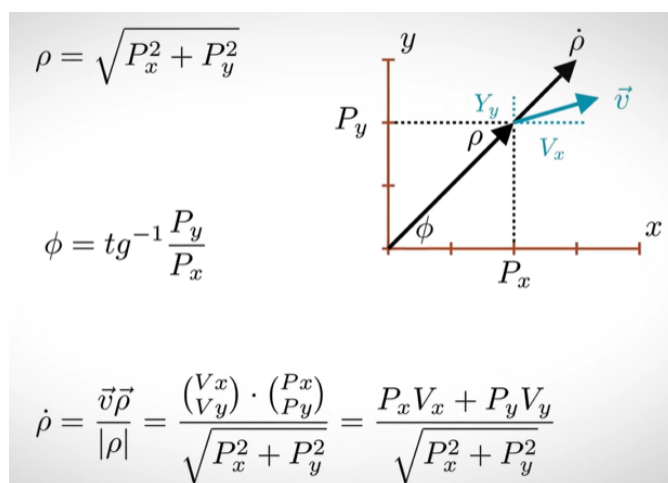
מה מקבלים מהחיישן?

1. **מרחק** - את המרחק מהחיישן לאובייקט שעליו אנחנו עוקבים. למרחק נקרא Rho ונסמן ρ .
2. **זווית** - כלומר הזווית בין החיישן ובין האובייקט. לזווית נקרא Phi ונסמן ϕ .
3. **מהירות** - כלומר את המהירות הרדיאלית - מדובר על המהירות שבה האובייקט מתרחק או מתקרב אלינו. למהירות נקרא Rho Dot ונסמן $\dot{\rho}$.

בחיישן הזה יש חוסר לינאריות, אין לנו שום קשר בין וקטור המצב לבין מרחב החיישן, כלומר אין לנו איך לחבר אותם יחד בהעתקה לינארית.



לאחר כל מיני ניתוחים מתמטיים ונוסחאות נקבל את הפרמטרים שלנו לחיישן *RADAR*:



ומכאן נקבל:

$$\hat{z}_t = \begin{pmatrix} \rho \\ \phi \\ \dot{\rho} \end{pmatrix} = \begin{pmatrix} \sqrt{P_x^2 + P_y^2} \\ \text{tg}^{-1} \frac{P_y}{P_x} \\ \frac{\begin{pmatrix} V_x \\ V_y \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \end{pmatrix}}{\sqrt{P_x^2 + P_y^2}} \end{pmatrix}$$

נשים לב שאף אחד מהמשוואות האלו לא לינאריות.

לינאריות במסנן קלמן מורחב

מבוא

התפלגות גאוסית שעברה טרנספורמציה לינארית היא עדיין התפלגות גאוסית וזו החוזקה הגדולה של התפלגות גאוסית וזו החוזקה הגדולה של מסנן קלמן. אבל לא תמיד אנחנו נוכל לייצר טרנספורמציות לינאריות. לפעמים המודל שלנו יהיה לא לינארי, לפעמים החיישנים יהיו לא לינאריים, ואז הדבר הזה כבר לא יעבוד. התפלגות שהיא לא גאוסית - אי אפשר להשתמש בה, זאת אומרת, אי אפשר להכפיל אותה בהתפלגות אחרת ולקבל התפלגות גאוסית. יש לנו כמה דרכים לטפל בסוגייה הזאת ובפרק הנוכחי, מסנן קלמן מורחב, או Extended Kalman Filter, או בקצרה EKF.

מסנן קלמן מורחב לוקח כל פונקציה אפילו לא לינארית ומנסה למצוא לה פונקציה קרובה שהיא לינארית על ידי **פיתוח טיילור מסדר ראשון**. זאת אומרת, מקרבים את הפונקציה על ידי פולינום ממעלה ראשונה למשהוא שהוא לינארי ומשתמשים בדבר הזה במקום בפונקציה המקורית. את ההכללה של פיתוח טיילור למימדים נוספים נראה באמצעות ייצור מתמטי שנקרא "יעקוביאן".

פיתוח טיילור

נקרב את הפונקציה על ידי פיתוח טיילור מסדר ראשון, כלומר, עבור פיתוח טיילור רגיל:

$$\sum_{i=0}^{\infty} \frac{f^{(i)}(a)}{i!} \cdot (x - a)^i$$

ניקח רק שני האיברים הראשונים:

$$f(a) + \frac{f'(a)}{1!} \cdot (x - a)^1$$

נוסחה:

$$f(x) = f(a) + \frac{f'(a)}{1!} (x - a) + \frac{f''(a)}{2!} (x - a)^2 + \frac{f'''(a)}{3!} (x - a)^3 + \dots$$

חסרון

החסרון במסנן קלמן EKF הוא שכאשר מתרחקים מ- a אז התוצאה פחות ופחות מדויקת. כי באמת מצד אחד מסנן EKF עובד עם הקירוב הלינארי אבל, כאשר המודל שלנו עובד בצורה דומה ללינארי, מסנן קלמן מורחב יעבוד טוב. אבל כאשר הוא מתרחק מהגבולות הלינאריים, מסנן קלמן מורחב מזייף. מסנן קלמן מורחב יעבוד טוב בסביבות הלינאריות.

מטריצת היעקוביאן

עכשיו, כאשר סיימנו עם פיתוח טיילור, ניתן לראות איך מכלילים את זה למימדים נוספים ומשתמשים בזה במסנן הקלמן שלנו.

$$\begin{aligned} X_{k+1} &= F \cdot X_k \rightarrow X_{k+1} = f(X_k) + r \\ Z_k &= H \cdot X_k \rightarrow Z_k = h(X_k) + w \end{aligned}$$

שתי הפונקציות האלה שאנחנו רואים פה, הן פונקציות שהן לא לינאריות ומה שאנחנו נצטרך לעשות זה לקחת את הפונקציות האלה שבעצם עובדות על כל וקטור המצב שלנו ולייצר מהן פונקציה לינארית. בטיילור אנחנו יודעים ש:

$$f(x) = f(a) + f'(a) \cdot (x - a)$$

את אותו הדבר אנחנו נעשה בכמה מימדים ואנחנו נעשה את זה ספציפית על מטריצת H .

$$h(x) = h(\mu) + \frac{d \cdot h(\mu)}{dx} \cdot (x - \mu)$$

אנחנו תמיד מפתחים את זה סביב הנקודה μ , שזאת הנקודה שיש בה את הכי הרבה סמפלים, מכיוון שהכל מתפלג בצורה גאוסית.

הרכיב $\frac{d \cdot h(\mu)}{dx}$ הוא בדיוק כמו הרכיב $f'(a)$ רק שלרכיב הזה אנחנו קוראים **מטריצת היעקוביאן**.

מטריצת יעקוביאן כללית, תראה כך:

$$\begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & & \\ \vdots & & \ddots & \\ \frac{\partial h_n}{\partial x_1} & & & \frac{\partial h_n}{\partial x_n} \end{bmatrix}$$

השורה הראשונה היא גזירת h_1 לפי x_1 בעמודה הראשונה, בעמודה השניה, וכו'.....

ניתן גם להסתכל לפי עמודה: שבעמודה ראשונה גוזרים את ה- h ים לפי x_1 , בעמודה השניה לפי x_2 וכו'...

דוגמה

נניח שיש לנו וקטור מצב רגיל של מיקום ומהירות, ווקטור חישן של RADAR שכפי שאמרנו נראים כך:

$$x = \begin{pmatrix} P_x \\ P_y \\ V_x \\ V_y \end{pmatrix}, z = \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} = \begin{pmatrix} \sqrt{p_x^2 + P_y^2} \\ \arctan\left(\frac{P_y}{P_x}\right) \\ \frac{\begin{pmatrix} V_x \\ V_y \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix}}{\sqrt{P_x^2 + P_y^2}} \end{pmatrix}$$

הבעיה היא ש-z בכלל לא לינארית - לכן נשתמש ביעקוביאן.

נסמן:

$$\begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} = \begin{pmatrix} \sqrt{p_x^2 + P_y^2} \\ tg^{-1} \frac{P_y}{P_x} \\ \frac{P_x \cdot V_x + P_y \cdot V_y}{\sqrt{P_x^2 + P_y^2}} \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} =$$

נגזור לפי כל פרמטר, לדוגמה נגזור לפי φ ונקבל:

$$\frac{\partial \varphi}{\partial P_x} = \frac{\partial \arctan\left(\frac{P_y}{P_x}\right)}{\partial P_x} = \frac{1}{\left(\frac{P_y}{P_x}\right)^2 + 1} \cdot \frac{-P_y}{P_x^2} = \frac{-P_y}{P_x^2 + P_y^2}$$

וכן הלאה, לבסוף נקבל את המטריצה הבאה:

$$\begin{bmatrix} \frac{P_x}{\sqrt{p_x^2 + P_y^2}} & \frac{P_y}{\sqrt{p_x^2 + P_y^2}} & 0 & 0 \\ -\frac{P_y}{\sqrt{p_x^2 + P_y^2}} & \frac{P_x}{\sqrt{p_x^2 + P_y^2}} & 0 & 0 \\ \frac{P_y(V_x P_y - V_y P_x)}{(P_x^2 + P_y^2)^{3/2}} & \frac{P_x(V_y P_x - V_x P_y)}{(P_x^2 + P_y^2)^{3/2}} & \frac{P_x}{\sqrt{P_x^2 + P_y^2}} & \frac{P_y}{\sqrt{P_x^2 + P_y^2}} \end{bmatrix}$$

דבר נוסף וחשוב מאוד זה המימדים של המטריצה.

שימו לב שהמימדים של המטריצה הם 3 על 4 והדבר הזה אומר לנו משהו חשוב לגבי מטריצת היעקוביאן.

למעשה מטריצת H זו המטריצה שמעבירה אותנו ממרחב המצב למרחב החישן.

מה שתראו בשאלות במבחן זה איך עושים את מטריצת היעקוביאן גם עבור פונקציית H אבל לא רק, גם עבור פונקציית F , גם עבור מודלים של תנועה שהם לאו דווקא לינאריים.

היתוך חיישנים במסנן קלמן מורחב

חיישנים שונים עובדים בצורה שונה ובזמנים שונים. ביחידה הזאת נדבר על היתוך חיישנים, וזה יוביל אותנו ליחידה הבאה בה ניקח את כל מה שלמדנו וניישם מסנן קלמן מורחב שלוקח מידע מחיישנים שונים ומייצר לי שערוך מיקום אופטימלי. אז מה זה בעצם היתוך חיישנים? - כאשר יש לי מידע מחיישנים שונים, השאלה היא איך אני יכול לקחת את כל המידע הזה ובעצם לחבר אותו ביחד כדי לקבל תוצאה שהיא יותר טובה מכל אחד מהחיישנים בנפרד?

נושאים של היתוך חיישנים

- **קליברציה** - כאשר מקבלים חיישנים, לא ניתן להתחיל לעבוד איתו מיידית וצריך להתאים אותו למערכת.
- **מיקום החיישן על הרכב** - כל חיישן ממוקם במקום מעט שונה בכל רכב, ולכן צריך לנרמל את כל החיישנים לנקודה אחת.
- **קצב הרענון** - כל חיישן נותן אינדיקציה בזמן שונה וצריך ליצור מנגנון שיודע לעבד את המידע בזמנים שונים.
- **אופן המדידה** - לרוב ניקח את המדידות המעובדות כלומר את הנתונים שהחיישן מקבל מתורגמים למידע שאנחנו נוכל לעבוד איתו.

החוזקה של היתוך חיישנים

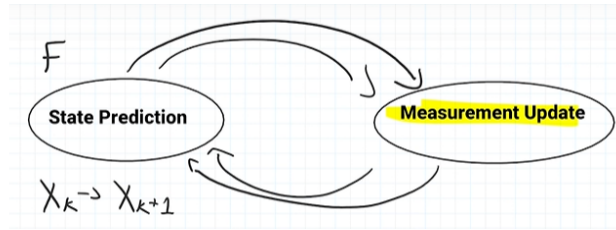
- היכולת לשלב חיישנים עם קצב עדכון נמוך ביחד עם קצב עדכון גבוה.
- כאשר אני עובד עם חיישן, יש לי שתי אופציות של מדידה שאני יכול לקחת:
1. **המדידות הגולמיות** - לחשב ולייצר את מה שהחיישן אמור לספק לי.
 2. **המדידות המעובדות** - כמו למשל מיקום הרכב או נקודות x, y, z .

מה עדיף? -

כאשר אני משתמש במדידות הגולמיות, בעצם אני מקבל משהו שהוא מצד אחד הרבה יותר מדויק אבל הרבה יותר קשה לעבודה מבחינה חישובית כי אני צריך לקחת את כל ענן הנקודות הזה ובעצם להוציא ממנו חישובי x, y, z שזה יכול להיות מאוד מאוד יקר מבחינה חישובית לרכב אוטונומי שגם עושה דברים אחרים. כדי לפתור את הבעיה הזאת, מה שאנחנו בדרך כלל מנסים לעשות זה להשתמש במה שנקרא **Loose Coupling**, שזה לקחת מהחיישן את המדידות המעובדות. המדידות האלה יהיו יותר יעילות מבחינה חישובית ומסנן הקלמן יפתור לנו את חוסר הדיוק.

מדידות בזמנים שונים

נחזור על מה שאנחנו עושים תמיד בכל מסנן בייסיאני ובייחוד בקלמן פילטר.



ה-Measurement Update זה לקבל את המידע מהחיישן, ולפי המידע מהחיישן, לקבל ניחוש שהוא טוב יותר, או מה שנקרא "ניחוש אפוסטריורי".

ה-State Prediction בא לידי ביטוי במטריצת F .

זה המודל הדינאמי של המערכת וזה מה שמביא אותי, או מעביר אותי, מ- X_k ל- X_{k+1} .

State Prediction מעלה את חוסר-הוודאות בעוד ש- Measurement Update מעלה את הוודאות, ולכן היינו רוצים כמה שיותר סשנים של Measurement Update.

והשאלה היא בדיוק זאת:

- איך אנחנו מתנהלים עם ה-Measurement Update כאשר יש לנו כמה חיישנים שונים?
- איך אנחנו עושים גם State Prediction וגם Measurement Update ביחד?

נזכר גם בחיישנים שלמדנו:

$$\begin{array}{cc} \text{radar} & \text{lidar} \\ \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} & \begin{pmatrix} P_x \\ P_y \end{pmatrix} \end{array}$$

מצב אסינכרוני במדידות בזמנים שונים

זה מצב שבו שני החיישנים לא מתואמים.

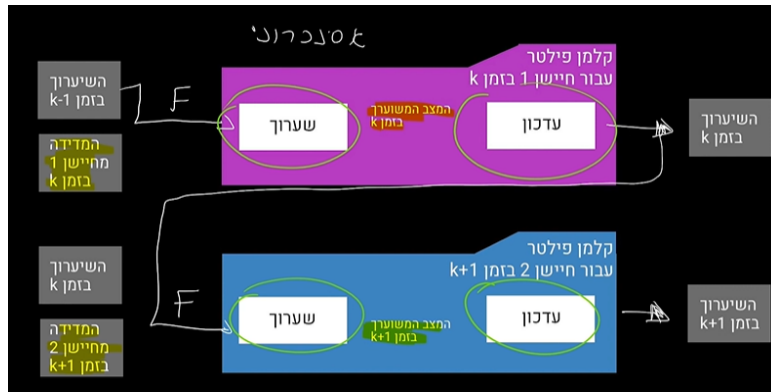
זאת אומרת, אני מקבל מידע מחיישן א' בתדירות מסוימת ואני מקבל מידע מחיישן ב' בתדירות אחרת, שהיא לא אמורה להיות מסונכרנת או מתואמת עם החיישן הראשון.

אז לצורך העניין, המדידה מהחיישן הראשון היא בזמן k , אנחנו לוקחים את איפה שהיינו בזמן $k-1$ ומשתמשים בפונקציית F כדי לשערך איפה אנחנו הולכים להיות בזמן k .
זה מה שנקרא "המצב המשוערך בזמן k ".

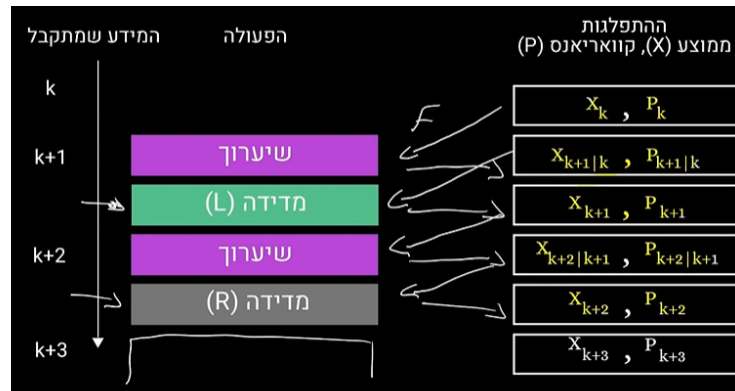
למעשה, עבור כל חיישן יש לנו 2 שלבים:

1. שלב השיערוך
2. שלב העדכון

כלומר, מגיע חיישן, אנחנו משערכים, או עושים Prediction, ואחרי זה אנחנו עושים Measurement Update, פעמיים, אחד לכל חיישן.



בכל זמן (k) אני מקבל מידה ומבצע עליה את השיערוך על פי האלגוריתם, ומבצע עדכון.



הסבר:

- נסמן את קבלת המדידה L כשלב $k + 1$ אז על פי האלגוריתם ניקח את המצב בזמן k ונבצע Prediction על ידי מטריצת F ואז באמת המידע והחיצוי יהיו בשלב $k + 1$ ומכאן נוכל לבצע את העדכון update.
- באותו אופן עבור המדידה R .

מה קורה למשל אם בנקודה $k + 3$ אני מקבל בו-זמנית גם מדידה מה-RADAR וגם מדידה מה-LiDAR? איך אז אני אתנהל כאשר אני ארצה להתיר את שני החיישנים ביחד? לכן נלמד גם על מדידה סינכרונית.

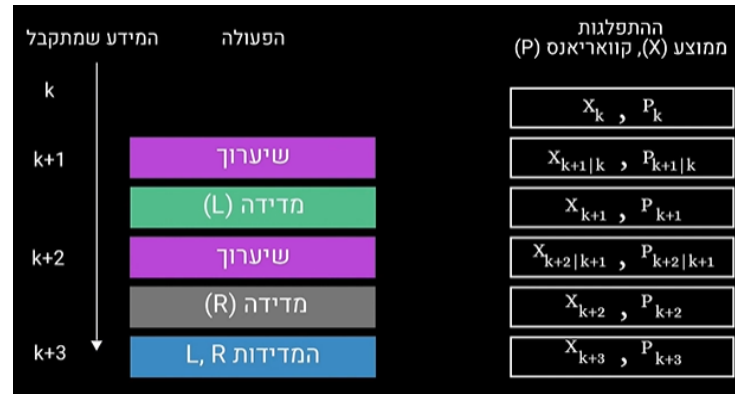
מצב סינכרוניים במדידות בזמנים עוקבים

במצב א-סינכרוני ביצענו לכל חישן בנפרד:

$Prediction \rightarrow Measurement Update$

במצב סינכרוני נפעל בצורה הבאה:

$Prediction \rightarrow L Update \rightarrow R Update$



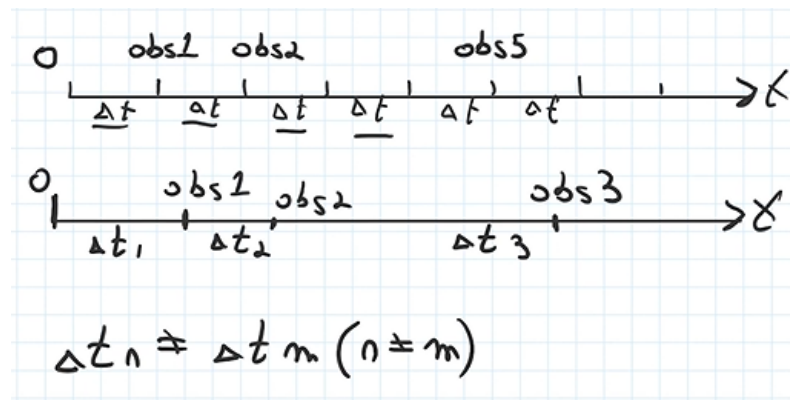
נזכיר ש-Prediction הוא הכפלה שלנו ב- F כלומר $X_{k+3} = F \cdot X_{k+2}$ ואין צורך לבצע אותו "פעמיים בנפרד" בזמן $k + 3$ ולכן נוכל לבצע אותם בצעד אחד.

זאת אומרת, אני יכול לעשות קודם כל עדכון לפי הרדאר ואז עדכון לפי הליידאר.

המעבר, פרידקציה, ממצב $k + 2$ למצב $k + 3$ בא רק כאשר אנחנו מקבלים את הסנסור ב- $k + 3$ ואז יש משמעות לפרידקציה, זאת אומרת, מה הייתי אמור לראות לעומת מה אני רואה בפועל.

מדידות בזמנים שונים

בחיים האמיתיים, לא נקבל את המדידות בהפרשי זמנים קבועים אלא שונים.



אנחנו רואים למשל בדוגמה כי כל שלב בפרק זמן שונה.

לכן, אנחנו צריכים בכל שלב לעדן את Δt ונקבל:

$$X = \begin{pmatrix} P_x \\ P_y \\ V_x \\ V_y \end{pmatrix}, F = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

שגיאת ה-RMSE

בשאלות של מבחן ובשאלות של קורס בדרך כלל יש לנו את הערך האמיתי, מה שגם נקרא ground truth. והערך הזה בעצם אומר מה בדיוק היה שם, איפה בדיוק היה הרכב, מה בדיוק הייתה המהירות שלו. אז יש כל מיני שיטות להעריך כמה האלגוריתם שלנו מדויק, והשיטה שאנחנו הולכים לעשות נקראת RMSE. root mean squared error. אתם תקבלו ווקטור של הערכים האמיתיים, אתם תקבלו את הווקטור של הערכים המשוערים ואתם תצטרכו לעשות את המשוואה הזאת. ברגע שיהיה לנו אותה אנחנו נדע עד כמה האלגוריתם שלנו מתקרב ונהיה יותר ויותר מדויק.

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (x_i^{Est} - x_i^{True})^2}$$

שתי החסרונות המשמעותיים במסנן קלמן המורחב :

- מטריצת היעקוביאן :
המטריצה מחשבת את הנגזרות החלקיות של הנתונים המגיעים מהמדדים לפי הפרמטרים המצויים בווקטור המצב של האובייקט ובדרך כלל החישובים הללו מסובכים ולא יעילים.
- לינאריזציה :
בתהליך הלינאריזציה אנו מנסים למצוא קירוב לינארי לפונקציה שאינה לינארית המתארת את התפלגות הנתונים אשר מגיעים מהחיישנים. ישנם מצבים שבהם הקירוב לא מצליח לתפוס את כל הנקודות במרחב המדידה ולכן במקרים מסויימים אלו - מסנן קלמן אינו יעיל כלומר אינו מספק את השערוך הטוב ביותר בהינתן המדידות הללו.

מסנן חלקיקים - Particle Filter

מבוא

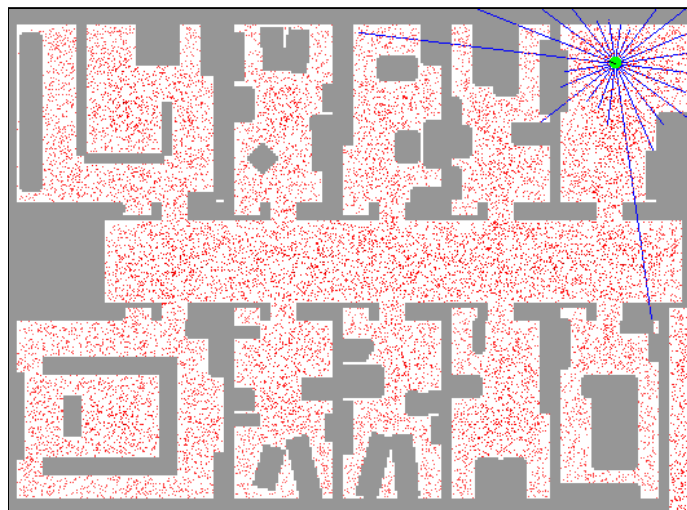
כאשר רובוט ממוקם ונדרש להציג לוקליזציה גלובלית, אנחנו צריכים לדעת היכן הוא נמצא בעזרת חישוב של חיישנים. כפי שאפשר לראות באנימציה מלמטה, הקווים הכחולים הם חיישני המרחק שמודדים את ההתפלגות האפרורית על היכן שהם נמצאים. הרובוט לא יודע היכן הוא מתחיל. בעצם, לגמרי לא בטוח היכן הוא נמצא כלל.

מסנן חלקיקים מציג שימוש בחלקיקים כפי שמוצג בנקודות האדומות באנימציה. כל חלקיק שכזה יש ניחוש יחיד עבור איפה הרובוט עלול להיות וכל חלקיק מאופיין בשלושת הערכים הבאים:

- קואורדינטה X
- קואורדינטה Y
- כיוון התנועה שלו (הנטייה שלו).

שלושת הערכים האלה מייצגים ניחוש יחיד. בעצם מאחר ויש לנו אלפי חלקיקים אז יש לנו אוסף של אלפי ניחושים במרחב וביחד הם יכולים לשער מיקומו של הרובוט.

תחילה, החלקיקים מפוזרים במרחב באופן אחיד ומסנן החלקיקים מאפשר להם להמשיך להתקיים במידה וחלקיק כלשהו הוא עקבי עם חישובי החיישן. כלומר, חלקיקים שעקביים יותר עם החישובים של החיישן סביר להניח שישרדו יותר והתוצאה מציבה מכך שבהסתברות גבוהה יאספו יותר חלקיקים עם הזמן וכך הם ייצגו יותר את האמונה של הרובוט. באופן מהיר מאוד, הרובוט מבין את צורת הפרוזדור בו הוא נמצא.



השוואה למסננים קודמים

• מסנן היסטוגרמה

- בדיד - לוקח את כל אזור ומחלק אותו לתאים נפרדים. אני יכול להיות בתא אחד או בתא שני או שלוש. ברגע שאני בתוך התא (bin) לא ניתן להבחין בין כל הנקודות בתוך התא. אם אנחנו רוצים להגדיל את הרזולוציה, אני חייב להקטין את גודל התא. לכן, מסנן ההיסטוגרמה הוא דיסקרטי (בדיד).
- מסוג Multimodal - אתה יכול להקצות הסתברות שונה לכל תא. אני לא מוגבל לניחוש אחד בלבד, אני יכול לצייר עם התאים, כל הפצה שאני רוצה.
- אקספוננציאלי - נגיד שאני במימד אחד ואזור העניין שלי הוא 100 מטר. רזולוציה של חד-מימד: 100^1 תאים.
רזולוציה של דו-מימד: $100^2 = 10,000$ תאים.
רזולוציה של תלת-מימד: $100^3 = 1M$ תאים.

• מסנן קלמן וקלמן מורחב

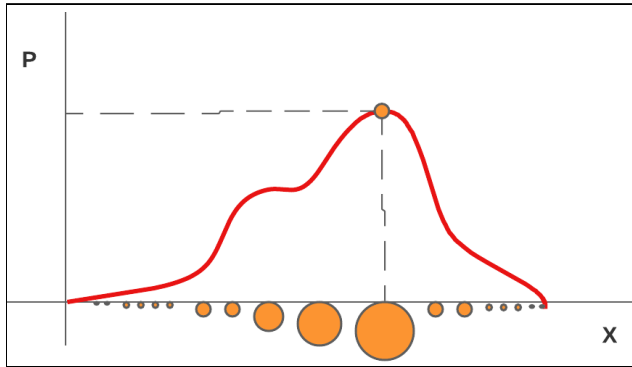
- רציף - מכיוון שפונקציית F מאפשרת לי לקחת את המצב הנוכחי ולהזיז אותה במילימטר קטן ימינה או שמאלה למצב הבא. מסנן קלמן מאפשר לי לזוז במרחב בצורה רציפה איך שבא לי ובדיוק אותו הדבר לגבי מסנן קלמן מורחב שמהבחינה הזאת הוא בדיוק כמו מסנן קלמן.
- מסוג Unimodal - קלמן פילטר יש לו התפלגות גאוסית ולכן יש רק ניחוש אחד שהוא μ כלומר וקטור המצב וזה ניחוש הטוב ביותר, ההערכה הטובה ביותר על מצב המערכת.
- ריבועי - בגלל שהרעיון הגאומטרי היא יעילות שהיא Quadratic, כאשר אני מוסיף מימד, אני מוסיף בסך הכול עוד שורה אחת במטריצה ועוד 2 שורות לוקטור המצב.

• מסנן חלקיקים

- רציף - מאפשר לי לקחת את ה-State הנוכחי ולהזיז אותו לאיזה כיוון שאני רוצה ובאיזו רזולוציה שאני רוצה. אני יכול לקחת כל חלקיק ולהזיז אותו לפי כל פונקציה.
- מסוג Multimodal - הוא לא מחייב לניחוש אחד. הוא יכול להחזיק ניחושים גם בצד ימין וגם בצד שמאל, בכל מקום במרחב המצב. הדבר הזה הוא משמעותי כי בעצם הוא לא מוגבל כמו מסנן קלמן.
- תלוי מקרה - ריבועי: אם אתה פורס את החלקיקים שלך בצורה נכונה ואם אתה מכוון את האלגוריתם שלך בתוך גבולות מסוימים לא תהיה מורכבות אקספוננציאלית.
- תלוי מקרה - אקספוננציאלי: אני צריך להוסיף חלקיקים ככל שאני מתקדם במימדים, נגיד שאני צריך 1000 חלקיקים לכסות אזור דו מימדי, אני אצטרך בערך מיליון חלקיקים לכסות את אותו אזור בתלת מימד.

	מרחב המצב	פונקציית האמונה	Efficiency
Histogram Filter	Discrete / בדיד	Multimodal	Exponentially
Kalman Filter	Continuous / רציף	Unimodal	ריבועי / Quadratic
Particle Filter	Continuous / רציף	Multimodal	תלוי מקרה

התפלגויות לא פרמטריות



נושא זה הוא הבסיס המתמטי למסנן חלקיקים. בהתפלגות גאוסית ראינו שבעזרת שני פרמטרים בלבד אפשר לצייר את ההתפלגות (פעמון) ולחשב אותה בכל נקודה במרחב המוגדר שלה. אבל מה קורה כאשר יש לנו התפלגות שרירותית שהיא לא מאופיינת כמו "הפעמון" שהכרנו בקלמן? יהיה לנו קשה למצוא את הפונקציה המתאימה להתפלגות הזאת ולבצע בה חישובים. לכן, במקרים כאלה אנחנו יכולים לתאר את

ההתפלגות השרירותית בצורה לא פרמטרית ע"י נקודות. בעצם על ציר ה- x נגדיר את המשקלים (גודל ההסתברות) בכל נקודה בפונקציה. הסתברות גבוהה - משקל גדול ובהתאם, הסתברות נמוכה במשקל קטן. ככה ניתן בעצם, על ידי חלקיקים, כאשר לכל נקודה אנחנו קוראים חלקיק. הגודל של החלקיק יתאר לנו את ההסתברות להיות באותו מקום לפי ההתפלגות השרירותית. כלומר, אנחנו מתארים את כל ההתפלגות השרירותית (אדום בגרף) בצורה שהיא לא פרמטרית עם חלקיקים. הדבר הזה מאפשר לנו בצורה פשוטה לתאר כל התפלגות שנרצה. ניתן לתאר את ההתפלגות השרירותית באמצעות נקודות בדיוק כמו שניתן לתאר את ההתפלגות הגאוסית באמצעות נקודות וזה המשמעות הגדולה של מסנן חלקיקים. כלומר, היכולת שלנו לתאר כל התפלגות שנרצה באמצעות נקודות (חלקיקים) ובצורה שהיא לא פרמטרית. הצורה המתמטית נראית כך:

$$\chi = \{(x^{[j]}, w^{[j]})\}_{j=1, \dots, J}$$

הסבר:

χ - מתאר לנו את אוסף כל החלקיקים, כאשר לכל חלקיק יש 2 פרמטרים:

1. $x^{[j]}$ - ההיפוטזה שלנו, בהקשר הקודם, זה ההיפוטזה שאני נמצא במקום מסוים, למשל $x = 8$.
2. $w^{[j]}$ - המשקל, כלומר החשיבות של הנקודה (החלקיק), ככל שהמשקל גדול ככה הנקודה גדולה יותר. במילים אחרות, ככל שהנקודה גדולה יותר זה אומר שיותר מסתבר שאנחנו נמצאים בנקודה הזאת.

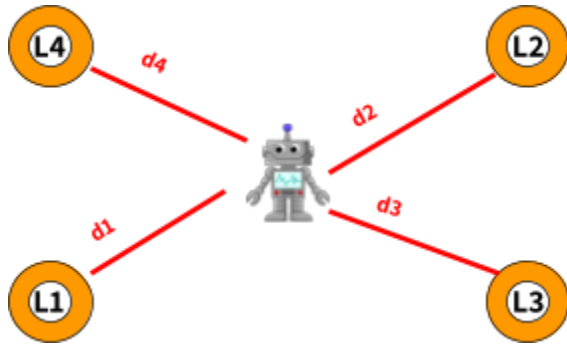
מרחב המדגם נראה כך:

$$\Pr(x) = \sum_{j=1}^J w^{[j]} \cdot \delta_{x^{[j]}}(x)$$

הסבר:

הפונקציה δ היא אותה פונקציה שבכל מקום שווה 0, חוץ ממקום אחד שבה היא שווה ל- ∞ . הגודל של δ בכל מקום הוא בעצם ה- w באותה נקודה j , כלומר ה- δ בונה לנו את השטח (או את הגובה) מציר ה- x לציר ה- y וככה עם אוסף של נקודות אפשר לתאר את פונקציית ההתפלגות שלנו. **הערה** - ככל שיהיו לנו יותר חלקיקים, כך ההתפלגות שנתאר תהיה יותר מדויקת.

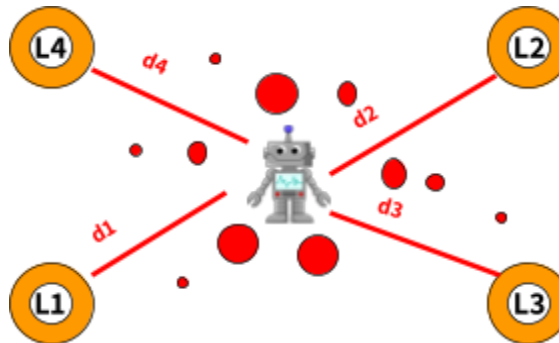
מרחב הרובוט



יש לנו 4 נקודות ציון (Landmarks).
והמרחקים מכילים את וקטור המדידה של הרובוט.
במרחב הזה נפזר באופן אחיד N חלקיקים כך שלכל
חלקיק מכיל $(x \ y \ \theta)$ כלומר מיקומו x, y וגם את
הנטייה שלו $\theta = orientation$.

משקלים

חוסר ההתאמה של המדידה בפועל והמדידה הצפויה מובילה אותנו לצורך במשקלים שאומרים לנו עד כמה
חשוב החלקיק, ובפרט ככל שהמשקל גדול יותר - ככה הוא חשוב יותר.
כעת, יש לנו הרבה חלקיקים ולכל אחד מהם יש משקל שונה ומוצגים לשם הדוגמה לפי גודל העיגול.

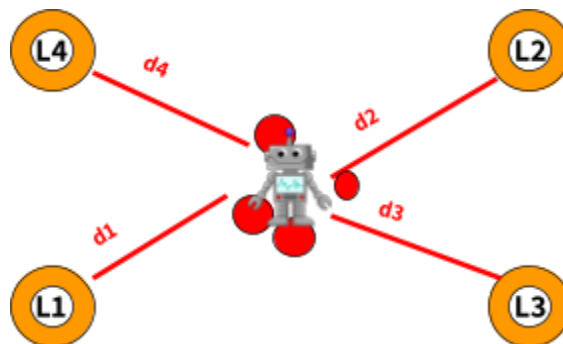


אנחנו נאפשר לחלקיקים האלה לשרוד באופן מקרי כלשהו אך ההסתברות של החלקיקים להישרדות תהיה
תלויה במשקלים שלהם.

למשל, באיור למעלה, חלקיקים עם עיגולים גדולים יותר ישרדו יותר מאשר חלקיקים בעיגולים קטנים.
לאחר מכן יש תהליך שנקרא **Resampling** שבעצם אחראי להציב N חלקיקים חדשים באופן אקראי
מהחלקיקים הישנים בהחלפה בהתאם לגודל המשקלים שלהם.

לאחר תהליך ה-Resampling, החלקיקים עם המשקלים הגדולים סביר להניח שישרדו יותר זמן בעוד
שהחלקיקים עם המשקלים הקטנים פשוט ימחקו.

כלומר, החלקיקים שמאוד עקביים עם מדידות החישה ישרדו יותר בהסתברות גבוהה והאחרים פשוט ימחקו.



דגימה חוזרת - Resampling

בהינתן N חלקיקים כאשר לכל חלקיק p_i יש שלושה ערכים $p_i = \{x_i, y_i, \theta_i\}$ וראינו שכעת יש להם גם משקלים w שהם בעצם מיוצגים בערכי *floats* או ערכים עקביים.

$$W = \sum_{i=1}^N w_i \quad \text{נקרא ל-} W \text{ סכום המשקלים של כל החלקיקים כלומר}$$

ננרמל את המשקלים לצורה הסתברותית ונסמן את המשקלים המנורמלים בתור α .

$$\alpha_i = \frac{w_i}{W} \quad \text{ונשים לב ש: } \sum_{i=1}^N \alpha_i = 1, \quad 1 \leq i \leq N$$

כעת, מה שתהליך ה-Resampling עושה הוא שהוא שם את כל החלקיקים עם המשקלים המנורמלים בתוך "שק אחד גדול" ואז הוא שולף מהשק עם חזרות, N חלקיקים חדשים לפי בחירת כל חלקיק עם ההסתברות α_i .

כלומר, החלקיקים עם ההסתברות הגבוהה ביותר יהיו חלק מהחלקיקים החדשים. אנחנו ניצור N חלקיקים חדשים כאשר יכולים להיות גם כמה העתקים זהים לחלוטין של חלקיק יחיד מהשק ששלפנו.

לדוגמה:

נתונים לנו 5 חלקיקים $\{p_i\}_{i=1}^5$ עם המשקלים של כל אחד בהתאם:

$$w_1 = 0.6, \quad w_2 = 1.2, \quad w_3 = 2.4, \quad w_4 = 0.6, \quad w_5 = 1.2$$

נרצה לדעת את ההסתברות של כל חלקיק p_i .

$$W = \sum_{i=1}^5 w_i = 6 \quad \text{נחשב:}$$

לכל $1 \leq i \leq 5$ נחשב $\alpha_i = \Pr(p_i) = \frac{w_i}{W}$ ונקבל:

$$\alpha_1 = 0.1, \quad \alpha_2 = 0.2, \quad \alpha_3 = 0.4, \quad \alpha_4 = 0.1, \quad \alpha_5 = 0.2$$

שאלה:

לפי הדוגמה הקודמת:

- האם ייתכן שחלקיק p_1 לעולם לא יידגם מחדש? - כן, לפי תהליך ה-Resampling.
- האם ייתכן שחלקיק p_3 לעולם לא יידגם מחדש? - כן, למרות שמשקלו גבוהה, עדיין יכול להיות מקרה שנבחר הארבעה האחרים, כיוון שמדובר כאן בעולם הסתברותי.
- לכן נשאלת השאלה, מהי ההסתברות שחלקיק p_3 לעולם לא יידגם שוב מחדש? והתשובה היא שתמיד נרצה לבחור את הארבעה האחרים ולא את p_3 , ההסתברות לכל הארבעה היא $0.6 = (1 - 0.4)$ לבחירתם, ולכן עבור 5 דגימות ב"ת נקבל: $0.0777 = (0.6)^5$.

גלגל ה-Resampling

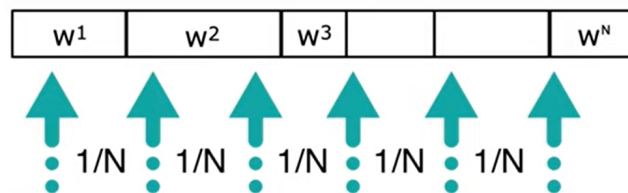
נציג את כל החלקיקים והמשקלים בגלגל רולטה גדול. כל חלקיק שולט על חתיכה בהתאם למשקלו. חלקיקים עם משקלים גדולים יותר יתפסו חתיכות עם נפח גדול יותר. במרכז הרולטה יש לנו חץ מסתובב, אם נסובב אותו, סביר להניח שהחץ יעצר ויצביע על חתיכות עם נפח גדול יותר, כלומר הוא ייצג באמת את הבחירה ההסתברותית לבחור חלקיקים עם משקלים גדולים יותר. הצגנו את נושא הגלגל כדי להסביר באופן קל ואינטואיטיבי את סיבוכיות תהליך ה-Resampling. בעצם אפשר להגיד שנחזור על התהליך הזה N פעמים, לכן זמן הריצה הוא לינארי, וכאשר החץ מצביע במקום מקרי, צריך להסיק איזה חלקיק הוא מצביע. כלומר, אנחנו צריכים לחפש בין כל החלקיקים עבור איזה חלקיק החץ מצביע בהתאם למשקל, את החיפוש אפשר לעשות בעזרת חיפוש בינארי שזה $O(\log N)$ מה שאומר שזמן הריצה של גלגל הרולטה הזה הוא $O(N \log N)$.

שיטת Low Variance Resampling

שיטת הגלגל תמיד משנה את התפלגות החלקיקים, ברוב הפעמים זה מעולה, אך לפעמים לכל החלקיקים יש את אותו המשקל. זה יכול לקרות מאי קבלת מידע מהחיישן או כאשר נרצה לעקוב אחרי 2 מקומות במקביל. למשל:

נדמין שיש לנו 2 חדרים שונים A , B , ונרצה לתחזק את העקיבה על החדרים האלה. יש לנו 10 חלקיקים בחדר A ו-10 חלקיקים בחדר B . אם נשתמש בשיטת גלגל הרולטה ל-Resampling: בשלב הראשון ייתכן שנקבל יותר חלקיקים מחדר B . בשלב הבא נקבל עוד ועוד חלקיקים מחדר B ובסופו של דבר באופן מאוד מהיר, אנחנו נפטר מכל החלקיקים שהיו בחדר A כי הם הצטרפו לכל החלקיקים בחדר B . כלומר, העלמנו ניחושים טובים מאוד בגלל חוסר מזל.

בעצם נרצה שיטה שבה אם לכל החלקיקים יש את אותו המשקל, אנחנו נסיים את תהליך ה-Resampling כאשר לא העלמנו אף חלקיק. כלומר, אנחנו לא נרצה לשנות את ההתפלגות. בשיטת Low Variance אנחנו לא בוחרים רק חלקיק אחד בכל פעם, אלא בוחרים בו-זמנית N חלקיקים. אפשר לדמין את זה בצורה של גלגל רולטה רק שהפעם יש לנו N חצים כך שהרווחים של החצים אחד מהשני שווים בגודלם. אם נסובב את הגלגל, כאשר הוא יעצר נדע באופן ישיר איזה חלקיקים צריכים לקחת. לכן, אם לכל החלקיקים יש משקל זהה, השיטה הזאת תתן לנו בדיוק אותם חלקיקים שהיו לנו.



אנחנו נבחר מספר באופן מקרי בין 0 ל- $\frac{1}{N}$.

עבור המספר שבחרנו, אנחנו נדלג בקפיצות של $\frac{1}{N}$ וכך נבחר כל חלקיק באופן הזה N פעמים. בסוף התהליך יהיה לנו N חלקיקים חדשים.

w^1	w^1+w^2	$w^{1:3}$			1
-------	-----------	-----------	--	--	---

$$\sum_{j=1}^N W_j = 1$$

for $j = 1$ to N :

$$U = r + j \times \frac{1}{N}$$

While $U > cum[i]$ $i++$

Pick Particle i

תרחיש Kidnapped Robot

כל החלקיקים נמצאים במיקום מסוים (מתרכזים בו) ומסונכרנים עם ה-Landmarks. הרובוט זז ומקבל דגימות אחרות לגמרי ואז כל החלקיקים הקודמים יהיו פחות רלוונטים והמשקלים ירדו. בחלק מהמקרים יכול להיות מצב בו הרובוט יעלם לנו בגלל מצב כזה. כדי שזה לא יקרה, בכל שלב נפזר כ-10% מהחלקיקים במקומות מקריים ואז נקבל מצב בהסתברות גבוהה יותר שלא נאבד את הרובוט (כמו תצפיתנים).

מודל Yaw Rate

נציג מודל תנועה כאשר $\dot{\theta} = \omega$ א:

$$x_f = x_i + v \cdot \Delta t \cdot \cos(\theta_0)$$

$$y_f = y_i + v \cdot \Delta t \cdot \sin(\theta_0)$$

לא ניתן לתאר תנועה במעגל בתרחיש זה. לכן, נציג מודל אחר בשם *Yaw Rate*. מודל זה מאפשר לתאר תנועה במעגל או בפניות ימינה/שמאלה. הזווית או האוריינטציה של הרכב משתנה בזמן, זאת אומרת, יש איזשהו שינוי של הזווית או של האוריינטציה לאורך הזמן. הפונקציה מקבלת חלקיק במיקום מסוים ועם זווית מסוימת ומזיזה אותו לפי המשוואות. בנוסף לחלקיק, הפרמטרים שהפונקציה מקבלת הינם המהירות והשינוי בזווית. עבורו $\dot{\theta} \neq \omega$ א:

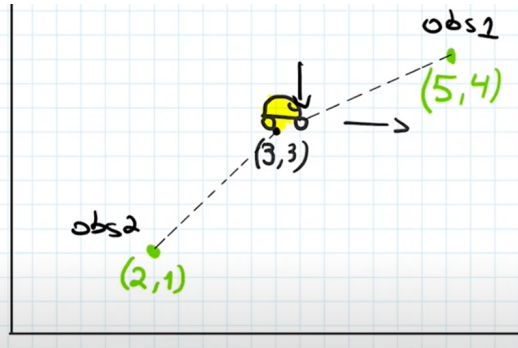
$$\theta_f = \theta_0 + \Delta t \cdot \dot{\theta}$$

$$x_f = x_i + \frac{v}{\dot{\theta}} \left(\sin(\theta_0 + \Delta t \cdot \dot{\theta}) - \sin(\theta_0) \right)$$

$$y_f = y_i + \frac{v}{\dot{\theta}} \left(\cos(\theta_0) - \cos(\theta_0 + \Delta t \cdot \dot{\theta}) \right)$$

התמרות הומוגניות

שיטה שמעבירה מהקורדינטות של הרכב שלי לקואורדינטות של העולם. מה שבאמת נרצה לראות זה איפה נמצאות הקואורדינטות האלה על המפה. נניח שהרובוט עומד בנקודה (3, 3), אנחנו מקבלים את הציפיות לגבי ה-Landmark בקואורדינטות שהם יחסיות לרובוט. החישוב, שנמצא ברובוט, מודד איפה נמצא ה-Landmark יחסית לקואורדינטות של הרכב.



במילים אחרות, $obs1$ היא בקואורדינטה (2, 1) יחסית לרכב. ו- $obs2$ היא בקואורדינטה (-1, -2) יחסית לרכב. כעת אנחנו יודעים היכן נמצאים ה-Landmark בקואורדינטות של הרכב שלנו. לצורך העניין, שהמיקום של הרכב הוא בנקודה (0, 0). אך מה שבאמת נרצה לראות זה איפה נמצאות הקואורדינטות האלה על המפה, כלומר איפה שבעולם האמיתי הם נמצאים. כדי לעשות את זה נשתמש ב-Rotation Matrix שבעצם מזיזה לנו את הקואורדינטות וגם משנה את המיקום שלנו. המטריצה נראית כך:

$$\begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & x_p \\ \sin \theta & \cos \theta & y_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}$$

הסבר:

- x_p, y_p הם הקואורדינטות של החלקיק.
- x_c, y_c הם הקואורדינטות של ה-Landmark.
- x_m, y_m הם הקואורדינטות של ה-Landmark במפה.
- הזווית θ מאפשרת לנו לחשב גם כאשר הרובוט בזווית ביחס ל-Landmark.

לסיכום, המטריצה מתרגמת את כל הנתונים למפה כך שהרובוט יעמוד בקואורדינטה (0, 0).

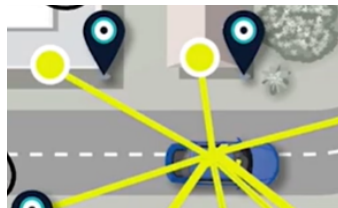
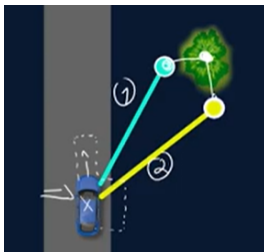
פונקציית משקל מתקדמת

בחישוב המשקל הנוכחי נכניס גם את המשקל של השלב הקודם - הסיבה שזו משפר את התוצאה כי ישנם חלקיקים שבשלב k נכניס לאזור "לא מוגדר" למשל קיר ובכך נגרמת אי-רציפות שמשפיע על התוצאות וגורמת ל"טעויות". הרבה פעמים בבעיות ניווט או לוקליזציה יש לנו חוסר רציפות, זאת אומרת, חלקיק שהיה יכול להיות טוב בנקודה מסוימת, ועכשיו הוא לא טוב, או שהוא נכנס בקיר, או שהוא יצא מהטווח או כל מיני דברים כאלה. כאשר פונקציית המשקל שלנו מתעלמת מהעבר, אי-הרציפויות האלה יכולות לגרום לטעויות גדולות. כאשר אני מסתכל גם על המשקל של החלקיק בפעם הקודמת, אני מוריד לי את טווח השגיאה. כלומר צריך לתת לו "עוד צ'אנס" שלא רק מכיל את נקודת הזמן הנוכחית שיכול לקבל בה טעויות אלא לבחון את החלקיק במצב הקודם שאולי כן היה טוב ואז ניתן לו צ'אנס. לכן נציג את *Bayesian Weight Function* המוגדרת כך:

$$weight(x_t^{[K]}) = k \cdot sense(x_t^{[K]}) + (1 - k) \cdot weight(x_{t-1}^{[K]})$$

Data Association

כאשר דיברנו על Landmarks (נקודות ציון) אנחנו יודעים איפה הם נמצאים. לפעמים ה-Landmarks לא נדע מה מיקומם, אנחנו נדע לזהות אותם אבל לא נדע מה המיקום שלהם. גם לא נדע להבדיל בין נקודות הציון x, y מה שכן נדע מה-Landmarks זה הזווית שלו מאיפה אנחנו נמצאים. אנחנו נרצה לקשר בין ה-Landmarks לבין המדידה שלנו. כלומר, לקשר בין נקודות הציון לבין כל מה שנמצא במפה מבחינה גלובלית. בציור משמאל, ההחלטה האם לבחור את 1 או את 2 בתור נקודות הציון שלנו היא החלטה משמעותית כי אחרי זה, אנחנו מגדירים את המשקל לחלקיק מסוים בדיוק לפי נקודות הציון האלה. אחת השיטות המאפשרות לנו לקשר בין נקודות הציון לבין המדידות שלנו היא שיטה שנקראת *Nearest Neighbor* שבעצם לוקחת את נקודת הציון שהכי קרובה למדידה שלנו. כלומר, ככל שהמדידה יותר רחוקה מנקודת הציון, ככה ההסתברות קטנה יותר.



חישובי שגיאה במסנן חלקיקים

עד כמה המסנן שלנו מתרחק או מתקרב לערך האמיתי? אנחנו צריכים להבין מה זה הערך האמיתי או ה-Ground Truth. הערך הזה בא לידי ביטוי בשלושה פרמטרים: x, y, θ . זאת אומרת, מהו הערך האמיתי של הרובוט שלנו ב- x , ב- y ומהי האוריינטציה האמיתית שלו θ ?

$$GroundTruth = g = (x, y, \theta)$$

אפשרות ראשונה: אנחנו יכולים לחשב את השגיאה כאשר אנחנו לוקחים בחשבון את כל החלקיקים, כל אחד לפי המשקל שלו. הדבר הזה נותן לי איזושהי שגיאה כאשר היא לוקחת בחשבון את כל החלקיקים:

$$error = \frac{\sum_{i=1}^M w_i \sqrt{(P_i - g)^2}}{\sum_{i=1}^M w_i} = \frac{\sum_{i=1}^M w_i |(P_i - g)|}{\sum_{i=1}^M w_i}$$

אפשרות שניה (טובה יותר): אנחנו עוברים על כל החלקיקים, מוצאים את החלקיק שיש לו את המשקל הגדול ביותר ואז אותו בוחנים אל מול המציאות.

$$error = |P_{Best} - g|$$

ה-error הוא וקטור, כלומר:

$$error = \begin{pmatrix} P_x - g_x \\ P_y - g_y \\ P_\theta - g_\theta \end{pmatrix}$$

בעיות חיפוש

תכנון הבעיה

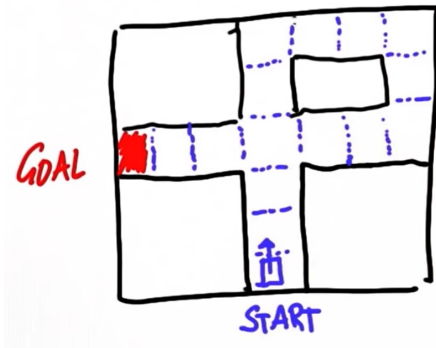
בהינתן:

- מפה
- מיקום התחלתי (מקור)
- מיקום סופי (יעד)
- עלות (משך הזמן)

המטרה היא למצוא את המסלול הזול ביותר בין המקור ליעד.

הפעולות:

- נסיעה ישרה
- סיבוב הרכב



בציור משמאל כל "קו מדורג" מגדיר צעד (כמו משבצת) וכל משבצת שכזאת מגדירה עלות של (+1). קל לראות באופן ראשוני שהמסלול יעלה בסה"כ 7. כלומר 6 פעולות של נסיעה ישרה ופעולה אחת של סיבוב הרכב.

חישוב עלות בשיטה שונה

כל מעבר ממשבצת למשבצת עולה לנו (+1) בשונה מהשיטה הקודמת בה כל משבצת בפני עצמה עלתה לנו 1. עלות המעבר משותפת גם לפניות ימינה ושמאלה וגם לנסיעה ישרה. לכן, לפי האיור משמאל, העלות הכוללת עכשיו תהיה 6.

חישוב עלות בשיטה שונה

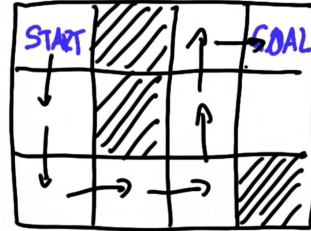
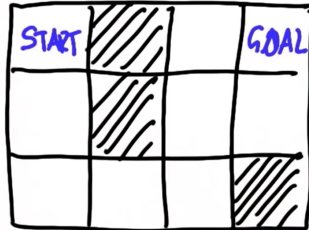
מה אם עכשיו העלות למעבר ממשבצת למשבצת עולה לנו (+1) מלבד מעבר בפנייה שמאלה בעלות (+10). העלות הכוללת של המסלול הזול ביותר עכשיו יהיה 15, כלומר המסלול נשאר אותו הדבר.

חישוב עלות בשיטה שונה

מה אם עכשיו העלות למעבר ממשבצת למשבצת עולה לנו (+1) מלבד מעבר בפנייה שמאלה בעלות (+20). העלות הכוללת של המסלול הזול ביותר עכשיו יהיה 16, כלומר המסלול השתנה, אנחנו צריכים לבצע סיבוב שלם מימין כדי להגיע אל היעד שלנו.

מבוך

נניח ויש לנו מבוך עם מקור ויעד, נרצה למצוא את המסלול הזול ביותר אל היעד.
נתון לנו רובוט עם יכולות פעולה (*up, down, right, left*).
לכן, המסלול הזול ביותר מהמקור אל היעד יעלה לנו 7.



בעיית חיפוש

ניקח את בעיית התכנון ונתייחס אליה כבעיית חיפוש.
נתון לנו רובוט עם יכולות פעולה (*up, down, right, left*).
תכנון המסלול או בעיית החיפוש היא מציאת סדרת הפעולות
הקצרה ביותר שתוביל את הרובוט מהמקור אל היעד.
מספר הפעולות שלנו במקרה זה הוא 11 פעולות.
● נלמד בקורס גם מימוש קוד לאלגוריתם לחיפוש מספר
פעולות הזול ביותר.

	0	1	2	3	4	5
0	S					
1						
2						
3						
4						G

	0	1	2	3	4	5
0	S					
1						
2						
3						
4						G

אלגוריתם חיפוש A*

אלגוריתם זה משתמש בפונקציה היוריסטית h .
לכל תא במפה שלנו יש ערך כלשהו $h(x, y)$.
באיור משמאל (הראשון) אפשר לראות שיש מכשול (משבצות
שבחור) שבהם אסור לדרוך.

הפונקציה היוריסטית h לא מתייחסת למכשולים שכאלה, היא יוצרת מפה

בגודל תואם למפה הקיימת ובכל תא יש ערך כלשהו המייצג עלויות.

לפונקציה היוריסטית חייב להיות ניחוש אופטימלי עבור עד כמה אנחנו

רחוקים מהיעד שלנו.

אכן אפשר לראות כי $h(0, 0) = 9$ ובאמת מספר הפעולות עד ליעד הוא 9.

	0	1	2	3	4	5
0	9	8	7	6	5	4
1	8	7	6	5	4	3
2	7	6	5	4	3	2
3	6	5	4	3	2	1
4	5	4	3	2	1	0

	0	1	2	3	4	5
0	S					
1						
2						
3						
4						G

נציג מקרה אחר כעת עם מכשול מורכב יותר.
מספר הפעולות הזול ביותר כעת הוא 13.
אבל כאן נכנס היופי של הפונקציה היוריסטית, היא לא חייבת להיות מדויקת.
כלומר היא מראה לנו את המרחק המשוער האופטימלי ביותר שניתן לקבל:

$$h(x, y) \leq \text{the true distance from } x \text{ to } y$$

כלומר הוא מציג לנו את החסם האופטימלי ביותר שניתן לקבל ללא התחשבות במכשולים כלל.
הפונקציה בעצם מראה לנו את המרחק האוקלידי אל היעד.

שלב התיקון

	0	1	2	3	4	5
0	S					
1	✓					
2	✓					
3	✓					
4	✓	✓				G

יש לנו רשימת פתיחה שמייצגת את המצב בו הרובוט נמצא בהתחלה
 $open[0, 0]$
נניח ואנחנו מתחילים $open[0, 0]$ ויורדים עד למטה לעבר $[4, 1]$:
נסמן ב- g את מספר הפעולות שנדרש ממנו להגיע עד למיקום הנוכחי.
נציג פונקצית חישוב f לתרחישים שכאלה:

$$f = g + h(x, y)$$

ובמקרה שלנו לפי האיור משמאל: $f = g + h(4, 1) = 5 + 4 = 9$
כעת נראה את היתרון של שימוש באלגוריתם ה- A^* :

	0	1	2	3	4	5
0	S					
1	✓					
2	✓					
3	✓		?			
4	✓	✓	✓	?		G

	0	1	2	3	4	5
0	9	8	7	6	5	4
1	8	7	6	5	4	3
2	7	6	5	4	3	2
3	6	5	4	3	2	1
4	5	4	3	2	1	0

אנחנו עומדים בשלב $[4, 2]$ ונרצה למצוא את המסלול הזול ביותר לכיוון G .
כלומר נרצה לדעת מה המשבצת הבאה שעלינו להתקדם אליה, לכן נחשב את f עבור 2 הפעולות האפשריות:

1. עבור $[3, 2]$ נקבל: $f = 7 + 4 = 11$.

2. עבור $[4, 3]$ נקבל $f = 7 + 2 = 9$.

ולכן עבור שתי הפעולות האפשריות שהצגנו נבחר את אפשרות (2) כיוון שהיא זולה יותר.

סרטון של A^* בזמן פעולה - [A* in Action - Artificial Intelligence for Robotics](#)

תכנות דינאמי

בהינתן:

- מפה
- יעדים

פלט:

- המסלול הזול ביותר מכל מקום במפה

למה שנרצה להתחשב במקרים של "כל מקום במפה"?
תכנות דינאמי נותן לנו אופציה אופטימלית לכל תא במפה שלנו.
נניח ורכב צריך לסוע ליעד מסוים בפנייה שמאלה, אך מחסום משטרתי חסם את הפניה שמאלה.
לכן, הרכב צריך להגיע אל היעד בדרך חלופית, לכן כדאי שנכיר "כל מקום במפה".

לכל תא יש תיוג (כמו סימון של חץ) שנקרא לעיתים $Policy$.
הפונקציה $Policy$ ממפה כל תא (משבצת) עם פעולה מסוימת כלומר $(x, y) \rightarrow Action$.
כל פעולה יכולה להיות $up, down, right, left$.
אנחנו נחשב את $Policy$ בעזרת תכנות דינאמי.

שיטת 3 הפעולות

נתעסק כעת במקרה של 3 מימדים (x, y, θ) .
כלומר, אם עד עכשיו התעסקנו ב-2 מימדים על (x, y) , כעת נתעסק עם מדד נוסף (x, y, θ) .
הסימן θ מגדיר לנו את הכיוון מהמשבצת (x, y) ולו הפעולות $up, down, right, left$.

שלושת הפעולות האפשריות לתנועה ממשבצת למשבצת הם:

1. התקדמות קדימה ממשבצת למשבצת
2. פניה שמאלה ורק אז התקדמות
3. פניה ימינה ורק אז התקדמות

אלגוריתם סופי בזמן פעולה Planning Conclusion - Artificial Intelligence for Robotics

סיכום

- העולם דיסקרטי.
- יש 2 אלגוריתמים לתכנון נסיעה:
 1. אלגוריתם A^* עבור מסלול
 2. אלגוריתם תכנות דינאמי - בשיטת $Policy$ על 3D

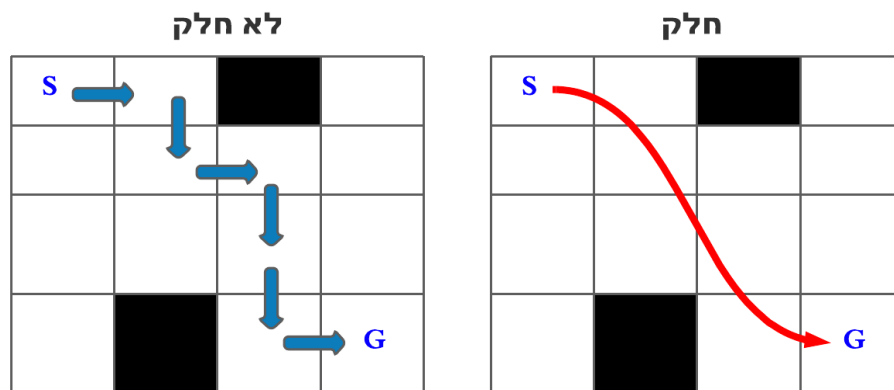
אלגוריתם PID

בנושא הקודם למדנו איך למצוא מסלול.
כעת נרצה להפוך את המסלולים האלה לפקודות של פעולות ממשיות.

אנחנו נדבר על:

- יצירת מסלולים חלקים.
- שיטת שליטה בשם PID .

ראינו שתכנון מסלול במפה דיסקרטית (עם משבצות מוגדרות) מכיל מסלולים דיי פשוטים וכמה מכשולים.
באיור למטה אפשר לראות את הכוונה במסלול חלק.
במציאות, רכב לא יכול לבצע פניות חדות כפי שמתואר במקרה ה-"לא חלק" שזה בעצם המקרים שנפגשו
בהם עד עכשיו בפרק הקודם.
בפרק זה נלמד ליצור מסלולים חלקים שיותר נכונים לתרחישי המציאות עם רכב.



אלגוריתם מסלול חלק

- תחילה ניצור משתנה y_i שהם בדיוק כמו כל ה- x_i כלומר $y_i = x_i$ כאשר x_i היא הפעולה ה- i מתוך סדרת הפעולות שיוצרות את המסלול הכולל.
- נרצה לשפר 2 קריטריונים:

a. נרצה למזער את הביטוי $(x_i - y_i)^2$.

b. נרצה למזער את הביטוי $(y_i - y_{i+1})^2$.

אם רק נשפר את (a) בלבד נקבל את המסלול המקורי כיוון שהוא מאותחל מראש $y_i = x_i$.

אם נשפר רק את (b) בלבד לא נקבל אף מסלול כיוון שהביטוי הזה לבד, אם הוא מתמזער אז כל ה-y-ים הם אותו הדבר מה שאומר שנקבל רק נקודה אחת (סדרת נקודות בגודל 1).

לכן בבירור ש-2 הקריטריות האלה בקונפליקט, במציאות אנחנו נמזער את שניהם בעזרת שימוש במשקל שנסמן אותו ב- α .

- ככל ש- α גדול יותר ככה המסלול יהיה חלק יותר.
- ככל ש- α קטן יותר ככה המסלול יהיה כמסלול המקורי עם הנקודות המקוריות (פניות חדות). לכן, אם נשפר גם את (a) וגם את (b) אז נקבל מסלול חלק.

אופטימיזציית הקריטריונים

איך נמזער את:

$$(a) \text{ נרצה למזער את הביטוי } (x_i - y_i)^2.$$

$$(b) \text{ נרצה למזער את הביטוי } (y_i - y_{i+1})^2.$$

אנחנו נמזער אותם בעזרת שימוש ב- **Gradient Descent**. בכל צעד ניקח את הצעד הקטן ביותר לכיוון שממזער את הטעות שלנו. כל איטרציה עבור 2 הקריטריונים מוגדרת כך:

$$y_i = y_i - \alpha(x_i - y_i)$$

$$y_i = y_i - \beta(y_{i+1} - y_i)$$

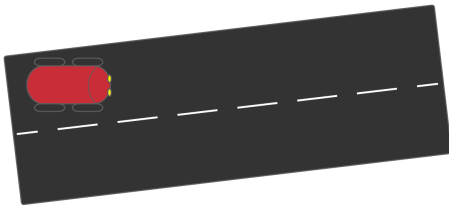
מיזוג הצעדים שמאלה וימינה מוגדרת כך:

$$y_i = y_i - \beta(y_{i+1} + y_{i-1} - 2y_i)$$

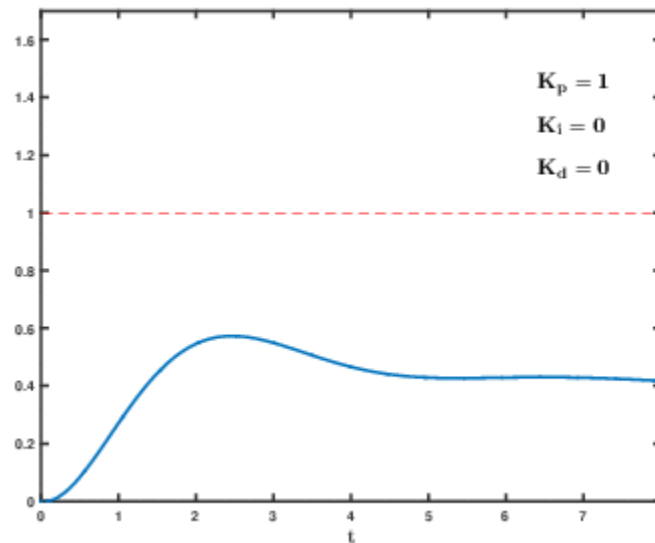
שאלה

אם נריץ את האלגוריתם עד לסיומו עם הפרמטרים הבאים $\alpha = 0$, $\beta = 0.1$, מה נקבל?
- אנחנו נקבל קו ישר בגלל שאנחנו לא משנים את הנקודה הראשונה והאחרונה במסלול.

אלגוריתם PID



ההשפעה של כל אחד ממרכיבי P , I , D של בקר ה-PID של ההיגוי. בקר ה-PID של ההיגוי מנתח את שגיאת המסלול הצולב (CTE) המסופק על ידי הסימולטור ומחשב את זווית ההיגוי שמנסה להחזיר את המכונית לקו המרכז (גודל מינימלי של CTE). לרכב יש "שגיאת מעבר" או בשפה רשמית CTE.



מבוא PID

P: בקרה פרופורציונלית P היא טכניקה פשוטה לכוון את הרובוט לפי ה-CTE, באופן פרופורציונלי. אם אנחנו רחוקים מקו הבסיס, הרכב שלנו יסטה לשם. אם אנחנו לא רחוקים מקו הבסיס שלנו, זה יעשה פניה עדינה. אמנם זה עובד, אבל אם נחשוב פעמיים על הבעיה, המכונית שלנו בעצם נעה סביב קו הבסיס שלנו כמו נהג שיכור וזה בהחלט התנהגות מסוכנת.

D: לפיכך, אנו רוצים להפחית את ההשפעה של התנודה. לכן אנו מציגים את הבקר השני, בקרת הנגזרת D . כאן, ניקח בחשבון את הסטייה שלנו, כדי להאט את המדרון והטעויות של P . נראה שבקרת PD מספיקה להיגוי. אבל, נכון? בואו נסתכל על התרחיש הבא: למכונית האוטונומית שלנו יש איזו בעיה מכנית שכאשר היא מתניעה, הגלגלים הקדמיים נסחפים קצת כמו למטה. אז כשהיא פועלת, התוכנה שלנו לא מודעת לבעיה שכן במקרה זה, עלינו לנווט יותר כדי לתקן את הגלגלים.

I: לפיכך, נראה את התוצאה הבאה. כדי להתמודד עם הבעיה הזו, נוסיף את האינטגרל I , כדי לגרום לרובוט לחשוב על כל התהליך ולתקן כל מקרה.

רכיב P

רכיב P, או "פרופורציונלי", מגדיר את זווית ההיגוי ביחס ל-CTE עם גורם פרופורציונלי τ .
 אנחנו מסובבים את הגלגלים בזווית פרופורציונלית: $\alpha = \tau + CTE$.
 למשל אם נשנה $0.3 \rightarrow 0.1 = \tau$ אז התנודה תהיה מהירה יותר.

לרכיב P יש את ההשפעה הישירה ביותר על התנהגות המכונית.
 זה גורם למכונית לנווט פרופורציונלי (ומנוגד) למרחק של המכונית ממרכז הנתיב (CTE) - אם המכונית רחוקה מימין היא תנווט חזק לשמאל, אם היא מעט חורגת בשמאל היא תנווט מעט לימין.

- ערכים גדולים יותר של P מביאים לתגובות מהירות יותר של זווית ההיגוי ביחס ל-CTE. כלומר, צריך להשתמש בערכים גבוהים יותר של P עבור מהירות גבוהה יותר של המכונית.
- ערכים קטנים יותר של P מביאים לתגובות איטיות יותר של זווית ההיגוי ביחס ל-CTE.

כעת נשאל מה יקרה לרכב עכשיו בעזרת הזווית הזאת? (לרכב בציור למעלה):
 הרכב יפספס את המטרה, הבעיה היא שלא משנה כמה קטן τ בסופו של דבר נסובב את הגלגלים קצת יותר מדי באופן חורג מהנתיב הנכון.

שאלה - האם יש דרך למנוע את מקרה (1) שהצגנו למעלה?

כן - בעזרת שיטה בשם **PD-Control**

רכיב D

הרכיב D, או ה"דיפרנציאלי", נוגד את הנטייה של רכיב ה-P לחרוג מהקו המרכזי.
 זה הרכיב הדיפרנציאלי של הבקר שעוזר לקחת נגזרת זמנית של שגיאה. משמעות הדבר היא שכאשר המכונית הסתובבה מספיק כדי להפחית את השגיאה, זה יעזור לא לחרוג דרך ציר ה-x.
 פרמטר D שמכוון נכון יגרום למכונית להתקרב לקו המרכז בצורה חלקה מבלי לחרוג בסיבובים.

- ערכים גבוהים יותר של P ידרשו גם ערכים גבוהים יותר של D.

במילים אחרות, הרכיב D נוגד את הנטייה של רכיב ה-P לחרוג מהקו המרכזי.

שליטת P+D

$$\alpha = -\tau_p \cdot CTE - \tau_d \cdot \left(\frac{d}{dt}\right) \cdot CTE$$

מה שזה אומר זה שברגע שרכב הסתובב מספיק כדי למזער את טעות CTE הוא יבין שהטעות מתמזערת כך שככל שהטעות קטנה עם הזמן, אז הרכב מתייצב בסיבובו לתיקון בעזרת $\tau_d \cdot \left(\frac{d}{dt}\right) \cdot CTE$.

הטיה שיטתית

לעיתים כשקונים רכב, קל לחשוב שהוא תקין והגלגלים בזווית מדויקת לכיוון הנכון. אך במציאות לפעמים הגלגלים קצת חורגים מכיוונם. עבור בני אדם זה לא ממש משמעותי כי אנחנו מטבענו תמיד מתקנים עם ההגה ללא הבחנה. במקרה הרבובטי, מקרים כאלה יגרמו לשגיאה חדשה CTE .

רכיב I

הרכיב I, או ה"אינטגרל", נוגד הטיה ב-CTE המונעת מהבקר P+D להגיע לקו האמצע. הטיה זו יכולה ללבוש מספר צורות, כגון הטיה שיטתית של היגוי. זה האינטגרל או סכום השגיאה להתמודד עם הטיות שיטתיות.

- רכיב ה-I גם משמש במיוחד להפחתת ה-CTE סביב עקומות.

מימוש PID

אם אנחנו נוסעים ברכב בכיוון ישר ומתרחקים מהמסלול התקין שלנו, שנתרחק מהמסלול עם הזמן אנחנו לא נוכל להתקרב למסלול. לכן, עם הזמן אנחנו נצטרך לשפר עוד ועוד את ההטייה של הרכב כדי לשמור על מסלול תקין. כדי לתאר את הרווח של השיפור מהרכב למסלול הנכון עם הזמן נסמן פורמלית:

$$\alpha = -\tau_p \cdot CTE - \tau_d \cdot \left(\frac{d}{dt}\right) \cdot CTE - \tau_I \sum CTE$$

- ככל ש: $\tau_I \sum CTE$ גדל עם הזמן הוא יתקן את התנועה של הרובוט לצורה הנכונה שלו.

נפרק את הנוסחה α :

- החלק $CTE \cdot \tau_p$ – מתאר את **P** (הפרופורציונליות)
- החלק $CTE \cdot \left(\frac{d}{dt}\right) \cdot \tau_d$ – מתאר את **D** כלומר תנאי השונות.
- החלק $\tau_I \sum CTE$ – מתאר את **I** כלומר אינטגרליות.

איך אפשר למצוא τ טוב מספיק?

בעזרת אלגוריתם Twiddle.

אלגוריתם Twiddle

באלגוריתם זה נרצה אוסף של פרמטרים.
 כדי לעשות את זה, הפונקציה שלנו `run()` חייב להחזיר לנו את ממוצע ה-*CTE*.
 נרצה לממש את האלגוריתם כדי למזער את ממוצע ה-*CTE*.
 במקרה כזה, הפלט של `run()` תלוי בשלושת הפרמטרים.

Twiddle Algorithm

```

1: Twiddle() :
2:     p = [0,0,0]
3:     dp = [1,1,1]
4:     best_error = run(p)
5:     while sum(dp) > 0.00001:
6:         for i in range(len(p)):
7:             p[i] += dp[i]
8:             error = run(p)
9:             if error < best_error:
10:                 best_error = error
11:                 dp[i] *= 1.1
12:             elif:
13:                 p[i] -= 2*dp[i]
14:             else:
15:                 p[i] += dp[i]
16:                 dp[i] *= 0.9

```

לכל קואורדינטה בפרט, נזיז את הפרמטרים שלנו למעלה או למטה לפי הערכים ב-*dp*.
 אם הוא מוצא פתרון טוב יותר, הוא שומר אותו ומעלה את הערך שלו ב-*dp*.
 אם הוא לא מוצא פתרון טוב יותר, הוא חוזר לפתרון המקורי ומוריד את הערך שלו ב-*dp*.
 אנחנו נבצע את זה שוב ושוב כל עוד הסכום של *dp* גדול יותר מהגבול.
 ברגע שהוא קטן מהגבול, זה אומר שאכן הגענו למתבקש.

הסבר קוד - Twiddle Algorithm

- 1:
- 2: ערכי ה-target שלנו שמאותחלים ב-0.
- 3: וקטור של שינויים פוטנציאליים שנרצה לחקור ונאתחל אותם לעת עתה עם 1.
- 4: נריץ את tux עם הפרמטרים p והוא יחזיר לנו את השגיאה הטובה ביותר שלנו עד עכשיו.
- 5: אנחנו נרצה לשנות את p כדי להקטין את השגיאה, לכן נריץ כמה פעמים עד שנגיע לסף המוגדר.
- 6: אנחנו עוברים ברצף על הפרמטרים האלה:
- 7: נרצה להגדיל את p בעזרת וקטור השינויים dp
- 8: לחשב שגיאה חדשה עבור ה- p החדש הזה
- 9: אם השגיאה החדשה הזאת טובה יותר מהשגיאה הטובה ביותר שיש לנו ביד אז:
- 10: נעדכן את השגיאה הטובה ביותר שלנו.
- 11: נשנה את וקטור השינוי שלנו לערך גדול יותר
- 12: אחרת, כלומר אם לא קיבלנו שגיאה חדשה שטובה יותר מהקיימת אז:
- 13: ננסה בדרך אחרת בשביל לשפר את המצב, אנחנו נחסיר פעמיים כי הוספנו כבר בשורה 7 וראינו כבר שזה לא עזר לנו במצב הנוכחי.
- 14: אם 2 התנאים נופלים אז:
- 15: נחזיר את p לערך המקורי שלו
- 16: ונקטין את וקטור השינוי שלנו

ניווט בתוך מבנים

(מתוך הרצאה משותפת עם המרצה בועז בן משה).

GPS עוזר לנו לדעת מיקום לא ידוע בכדור הארץ על בסיס מיקום ידוע בחלל (מיקום הלווין). זה נולד מהדבר ההפוך - ספוטיק.

טרנגולציה

רוצים לדעת איפה אובייקט נמצא על סמך זווית שיוצרים שלושה ביקונים - לוויין, wifi ועוד משהו. זה סנסור קלאסי שמחשב את המרחק על סמך זווית ההחזרה מאובייקט (כשהוא קרוב הזווית מסוימת וכשהוא רחוק הזווית אחרת).

טריאנגולציה

מחשב מרחק על סמך הזמן שלקח לו לקבל את הסיגנל ששלח בחזרה. הבעיה עם זה שאם המהירות גבוהה מידי, למשל מהירות האור, והחיישן שלנו מפספס במעט - נאמר מילי שנייה, הטעות תהיה גדולה מאוד (300 מטר). כלומר ככל שהמהירות גבוהה יותר צריך שיעון יותר מדויק (אבל זה לא באמת קורה במציאות). הפתרון: אם ניקח שני לוויינים לא מדויקים, הם יספיקו כדי לומר לנו איפה אנו נמצאים במימד אחד, בשני מימדים - ארבעה לוויינים, וזה מוגדר כמספר הלוויינים שצריך כדי לשערך מיקום, ולא שלושה (למרות שהמתמטיקה מספיקה גם איתם) כי יש את האלמנט של הזמן. כמובן שיש גבולות גזרה שאנו נמצאים על כדור הארץ.

מסלול גאוסטציונרי

לווייני GPS נמצאים כ-36 אלף ק"מ ממרכז כדור הארץ במסלול גאוסטציונרי הסיבה למרחק הזה היא שמהירות הסיבוב של הלווין מקביל למהירות הסיבוב של כדור הארץ.

ממד DOP

DOP הוא ממד לרמת הדיוק שלנו, אפשר גם לחלק את זה ל-2 סוגים:

1. $DOP - \sigma$ כלומר אי דיוק בצד הרוחבי.

2. $DOP - h$ כלומר אי דיוק בצד האופקי.

צריך לקחת לוויינים שהפירמידה שהם יוצרים היא בעלת הנפח הגדול ביותר, או DOP קטן. נתוני הגובה ב-GPS - הם אינהרנטית פחות מדויקים מנתוני ה-x,y.

Intersection Sphere

נניח שיש ארבעה לווניים שלכל לווין i יש נתוני מיקום (x_i, y_i, z_i) ואנחנו רוצים על סמך הנתונים האלו לחשב את המיקום של האובייקט, או של הuser (למשל ב-waze). יהיה לנו כאן סט של ארבע משוואות בארבעה נעלמים וזה ייתן לנו פתרון אחד.

--- להעתיק את המשוואות מהמצגת ---

בכל זאת waze משתמשים ביותר לווניים כדי להגיע לפתרון האופטימלי - זה שהמרחק בין הקווים יהיה מינימלי. מכל מקום בעולם בכל זמן ניתן לראות מעל האופק לפחות ארבעה לווניים. היום יש 32 לווניים.

לסיכום, GPS זו המערכת האמריקאית. GNSS זה של כולל לכל המערכות האלו (יש מערכות נוספות - למשל המערכת הרוסית נקראת GLONASS) כל מערכת עובדת בשיטת שידור אחרת.

SA Selective Availability

בהתחלה נתנו את ה-GPS - רק לאנשים שרצו (הצבא) ולאזרח הפשוט נתנו משהו לא מדויק בכלל שאי אפשר לעבוד איתו. לאחר מכן הייתה נקודה בזמן שמשם כבר נתנו לכולם להשתמש בזה בצורה מדויקת, ורק מאז היה אפשרי לעשות ניווט ברמה של לדעת לאיזה כיוון פונים (ימינה/שמאלה) וכו'...

GPS עובד בצורה של CGNA כלומר לכל לווין ה-GPS - יש אותו תדר. המערכת עצמה מורכבת. להרוס GPS זה קל - שמים רעש בתדר מסוים כך שכבר לא יהיה ניתן לקרוא אותו

Multipath

ייתכן שעצמים על פני כדור הארץ יסתירו את הלווין. לכן באזורים אורבניים הדיוק הרבה פחות טובות בגלל תופעות של Multipath וזו הבעיה המרכזית שם. אם יש שני חישנים שמחשבים את אותו הדבר ואחד דפוק בגלל שדות מגנטיים למשל, זה לא יעזור. לעומת זאת אם הם מחשבים את זה בצורה שונה זה יעזור.

אפקט דופלר

משהו שמתקרב אלינו ייקלט כאילו התדר יותר גבוה ומשהו שמתרחק ייקלט כתדר נמוך, למרות שהתדר הוא כל הזמן אותו הדבר. דוגמה טובה לכך היא אמבולנס עם סירנה שמתקרב אלינו ואז מתרחק, הסאונד משתנה עם המרחק. זה נקרא אפקט דופלר.

וויז עובד לפי קונספט בשם Matching Map וויז היא לא אפליקציית GPS, אלא היא אפליקציית ניווט שעושה חישובים מדויקים מנתוני GPS לא ממש מדויקים אשר מגיעים מהפלאפונים.

Matching Map אומר שהוויז "מדביקה" אותנו למפה איפה שיש מפה. זו הסיבה שלוקח הרבה זמן עד שוויז קולטת שיצאנו מכביש מסוים במקום להמשיך ישר כפי שהיא אמרה.

קואורדינטות

- **ECEF** - הכי פשוט אך הכי פחות טוב. ציר z הוא הציר מהקוטב הצפוני לקוטב הדרומי, x ו- y על קו המשווה. מסתמכים על כך שמרכז כדור הארץ הוא הנקודה $(0, 0, 0)$ ומשם מחשבים את המיקום לאורך לרוחב ולגובה. לכל אובייקט בכדור הארץ יהיה אותו ECEF בכל רגע נתון. הבעיה עם כך היא שלא נוח לעבוד עם x, y, z , האלו, כי z הוא לא בדיוק למעלה בקו ישר. לכן בד"כ עובדים עם קווי אורך וקווי רוחב.
- **Lat/Lon** - מחלקים את העולם לקווי אורך וקווי רוחב. קווי רוחב אומרים משהו משמעותי, תלוי גם בעונות השנה וכד', אך קווי האורך לא אומרים משהו משמעותי - סתם החליטו שקו אורך 0 יעבור בגריניץ. זה גם משפיע על השעות. קו אורך היה קשה לחישוב, כי צריך לצאת עם שעון מדויק מגריניץ ולראות מה השעה שם לעומת השעה שיש לנו לפי השמש, כי לא היו שעונים מדויקים. לאחר שיצרו את השעונים המדויקים כבר היה קל יותר לחשב אותם. גם כאן המספרים עובדים בזווית, כלומר לא מקבלים רק מספר במטרים מעל פני הים, אלא גם זווית. זה מה שהפלאפונים משתמשים בו בד"כ.
- **UTM** - מערכת קואורדינטות נוחה מאוד. כאשר יש $+1$ בציר x זה אומר שהתקדמנו מטר אחד צפונה, בציר y זה אומר מטר אחד מזרחה... הגובה בכל הנקודות כאן הוא ללא זווית, אלא במספר אמיתי.

פרוטוקולי GPS

1. **NMEA** - פרוטוקול טקסטואלי, מעין אסקי. יחסית איטי.
2. SiRF Binary to RINEX
3. Proprietary NMEA ST-Micro
4. Android API (LAT/ LON, COG/ SOG)

אפשר לעזור ל-GPS עם שלושה פרמטרים: **חסר**

דברים חשובים: איך GPS עובד (טרנגולציה, טריאנגולציה), מספר הקסם 4 (למה הוא חשוב), אופני הפעולה של GPS, החשיבות של חישוב נתונים שונים בצורה שונה - למשל לחשב את המהירות לבד ולא לגזור אותה מהמיקום.

ניווט בתוך מבנים

יש לכך כמה שיטות - אחת מהן היא wifi בעזרת מיפוי של הראוטרים. אבל זה לא תמיד עובד.

InDoor Positioning

רוצים לדעת איפה אנחנו נמצאים בתוך מבנה סגור (נניח קניון).
בסמארטפון יש כמה סנסורים שניתן להשתמש בהם:

- מצלמה - אבל זו בעיה בגלל בעיות של פרטיות וכד'.
- GPS - הבעיה בו היא שהוא לא עובד טוב בתוך מבנים.
- wifi - אבל צריך לדעת למפות את המיקומים של הראוטרים.
- סלולר - ריבוי אנטנות. הבעיה היא שגובה מושפע מאוד מהקליטה.
- בלוטוס -

לפעמים כשיש הרבה יותר משוואות מנעלמים זה פחות מדויק מאשר מספר קטן של משוואות.
דוגמה לכך היא שמכשיר שעוקב אחרי יותר מ 100- לווינים בו זמנית התגלה כפחות מדויק מאשר GPS רגיל (שמשמש בארבעה לווינים בלבד).
יש הרבה סנסורים והקושי הוא לחבר אותם. החיבור של giro (?).

שאלות

ראו את הפרק השני (sensors) [בקורס הדיגיטלי](#) של פרופ' בן משה וענו על השאלות הבאות.

שאלה 1 - איזה חיישנים אחראים להחלטה האם לפתוח כרית אוויר ברכב?

תשובה 1 - החיישנים האחראים להחלטה האם לפתוח כרית אוויר ברכב הם חיישנים משני סוגים חיישן תאונה אלקטרוני אשר מזהה מצבי תאונה על פי מצבים מוגדרים ולרוב יהיה כלול:

- חיישן שהוא מד תאוצה כאשר יש עליו תאוצה שלילית גבוהה אז הכריות אוויר ייפתחו.
- חיישן המודד את מהירות הגלגלים.
- חיישן המודד את מהירות הבלימה.
- חיישן הבודק האם המושב מאויש.

שאלה 2 - מה ההבדל בין חיישן תאוצה לבין חיישן תאוצה לינארית?

תשובה 2 - ההבדלים בין חיישני התאוצה:

- לא לינארי הוא שחיישן הכולל במדידות שלו את כוח הכבידה
- חיישן לינארי לא כולל את כוח הכבידה במדידות שלו.

שאלה 3 - מה עיקרון העבודה של חיישן optical-flow? מהי הרזולוציה הממוצעת שלו?

תשובה 3 - עקרון העבודה של החיישן Optical-Flow היא מדידה של מיקום לפי ניתוח של תמונות רבות שנלקחו ע"י החיישן וכך החיישן יודע להבדיל את שינוי המיקום שלו במרחב. הממוצע של הרזולוציה הוא 0.1 מילימטר.

שאלה 4 - מה מודד חיישן gyro? מה אומר האינטגרל שהוא יוצר?

תשובה 4 - החיישן מודד את האוריינטציה של המכשיר בו הוא נמצא, ואת המהירות הזוויתית שלו. האינטגרל של החיישן מספק לנו את האוריינטציה של המכשיר בו הוא נמצא, ואת המהירות הזוויתית שלו.

שאלה 5 - מנה את הבעיות בחיישן המודד שדה מגנטי.

תשובה 5 -

- שגיאות אשר מונעות שיערוך התנהגות מדויק.
- מיקום החיישן בגוף המכשיר חשוב, אחרת נקבל תוצאות שגויות.

שאלה 6 - איך לחץ ברומטרי עוזר בניווט בתוך מבנים?

תשובה 6 - לחץ ברומטרי עוזר בניווט בתוך מבנים כיוון שככל שעולים בגובה הלחץ הברומטרי קטן (ברזולוציה מאוד גבוהה – אפילו מילימטרים) ובכך נוכל לדעת את החיישן שלנו ביחס לנקודה מסוימת במרחב.

שאלה 7 - מה מיוחד בלווייני gps מהדור השלישי?

תשובה 7 -

- הלוויין מיועד לתוחלת חיים של 15 שנה.
- ביחס לדור הקודם מספק קליטה טובה יותר.
- נותן הספק יותר גבוה מהדורות הקודמים.

חלאס.