האוניברסיטה הפתוחה
המחלקה למתמטיקה ולמדעי המחשב

עבודה מסכמת בנושא
"עקיבה אחר מטרות מוטסות בעזרת מבני נתונים מרחביים"

עבודה מסכמת זו הוגשה כחלק מהדרישות לקבלת תואר
"מוסמך למדעים" .M.Sc במדעי המחשב
באוניברסיטה הפתוחה
החטיבה למדעי המחשב

על-ידי
דוד רוזנטור
306359985

העבודה הוכנה בהדרכתו של ד"ר ג'ק וינשטין

2011

תוכן עניינים

רשימת איורים

1 תקציר

מטרת המערכת שעוקבת אחר מטרות מוטסות היא לשערך את תנועת העצמים במרחב האווירי או בחלל, באזור עניין, וכך לתת מצע לקבלת ההחלטות לאנשים או למערכות שאחראיות על תנועה מוטסת באזור זה. מערכת העקיבה צריכה להתמודד עם מספר אתגרים: יכולת גילוי ואיכון של המטרות בסביבת מפריעים, שערוך מדויק של המסלול, שיוך נכון של המדידות למטרות כאשר יש ריבוי גילויים וכו'. מצד שני, מערכת כזאת צריכה להיות יעילה ולהתמודד עם מגבלות ואילוצי זמני תגובה; כמו כן, המערכת צריכה להיות זולה מספיק על מנת להיות כלכלית ותחרותית.

בדרך כלל, מערכת מורכבת ממספר שלבים שכל אחד מהם אחראי על חלק מהמשימה: חישנים אחראים על גילוי העצמים, תהליך האיכון מחשב את מיקומו של העצם והעוקב משערך את התנועה שלו. בגלל המגבלות הפיסיקליות לא ניתן לבצע אף אחד מהשלבים אלו בצורה מושלמת ולכן כל אחד מהתת-תהליכים אלו משתמש בשיטות סטטיסטיות לשערוך התוצאה.

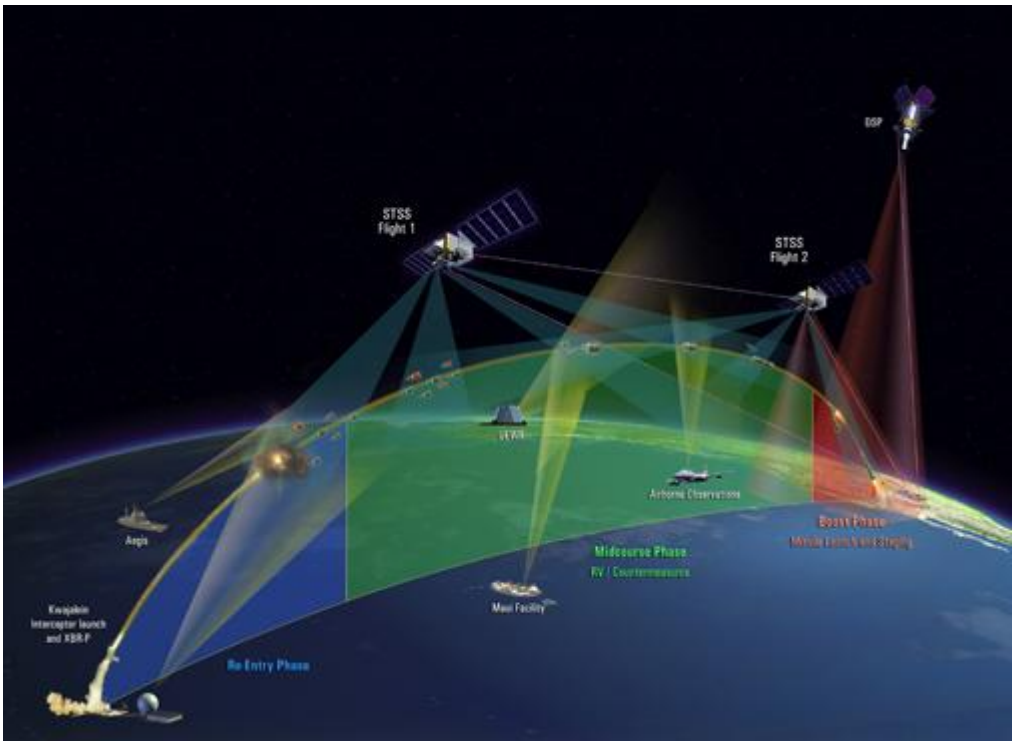
אחד המרכיבים החשובים הוא שיוך של המדידות שנמדדו ואוכנו ע"י המערכת למטרות שכבר מנוהלות. שיוך זה נעשה בעזרת חיפוש המועמד בעל התאמה מרבית בין כלל מדידות לכלל המטרות הידועות. ברור שיעילות החיפוש משפיעה על הביצועים של כלל המערכת. בעיית החיפוש היא בעיה נחקרת כבר שנים רבות. ישנן עבודות שחישבו חסמים שמכתיבים את הביצועים האופטימליים. בגאומטריה חישובית ישנו תחום שמטפל בבעיית החיפוש בעזרת מבני נתונים מרחביים. מבני נתונים אלו מאפשרים חיפוש יעיל במרחבים רב ממדיים ומהווים מאין אינדקס של חיפוש.

בעבודה זו תסקרנה שיטות איכון ועקיבה שהתפתחו במהלך שנים אחרונות. כמו כן, תוצג סקירה של שיטות חיפוש שונות במרחב חד-ממדי ודו-ממדי תוך כדי התייחסות לביצועים של השאילתות.

2 מבוא

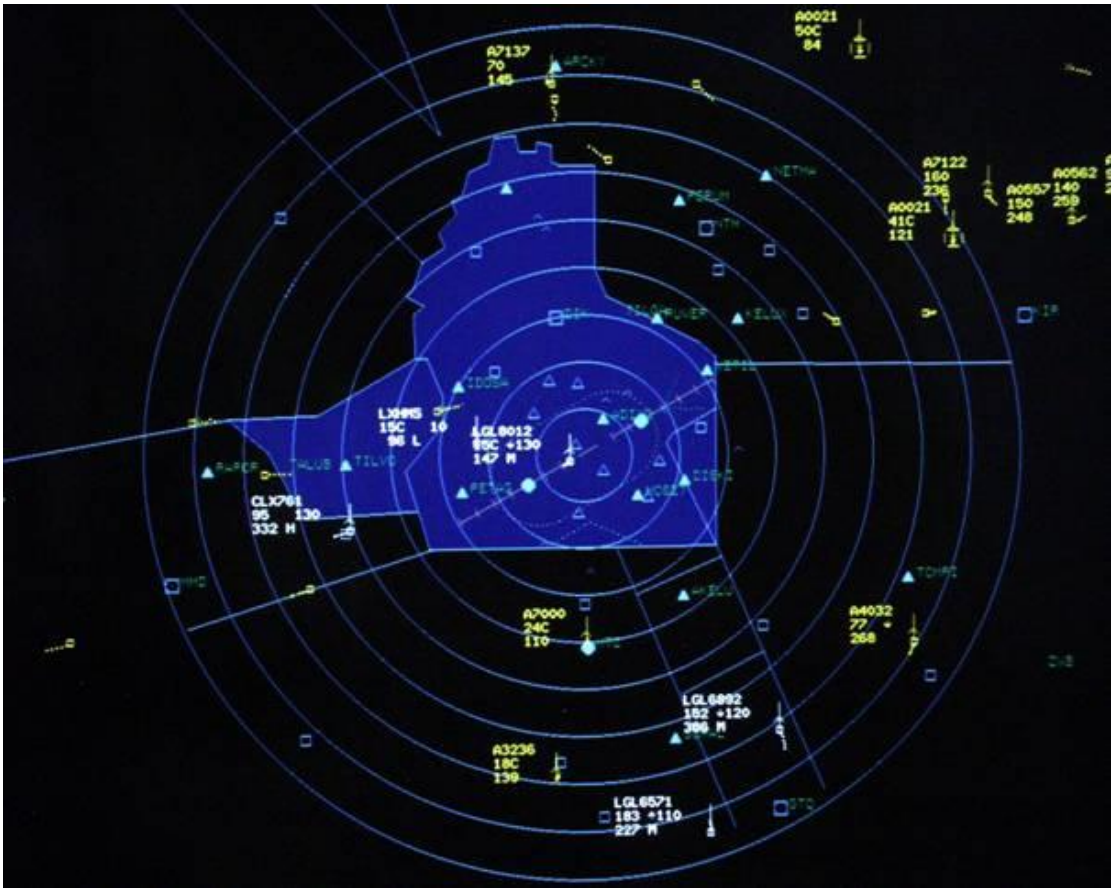
עקיבה אחרי מטרות היא אחת הבעיות הנחקרות מזה מספר עשורים אחרונים. בדרך כלל ה"עוקב" (tracker) – התהליך שמבצע את העקיבה, הוא חלק ממערכת גדולה עם מספר חישנים (sensors), לעתים מסוגים שונים. מערכות מסוג זה יכולות לשמש למטרות שונות גם בתחום הביטחוני וגם בתחום האזרחי.

דוגמא למערכת מהתחום הביטחוני: מערכת הגנה נגד הטילים. מטרת מערכת מסוג זה תהיה לסרוק את האזור ולנסות לזהות מטרות אפשריות. לאחר זיהוי העצם יש צורך לעקוב אחרי התנועה שלו על מנת להחליט האם הוא מהווה איום, כלומר האם הוא נע לאזור המוגן או לא. אם מחליטים ליירט את העצם המאיים אז משגרים טיל יירוט. הטיל הזה צריך לשנות את הכיוון שלו בהתאם לתנועות המטרה ולכן כרגע נדרש לעקוב אחרי שני עצמים: אחד – המטרה המקורית: צריך כל הזמן לשערך את המיקום העתידי שלו, מצד שני יש צורך לעקוב גם אחרי טיל היירוט: יש צורך לדעת מה מיקומו הנוכחי על מנת לשערך את וקטור התנועה האופטימלי ולתת פיקוד למנועים. דיוק העקיבה מחד, ומהירות תגובה לשינויים מאידך, הם התנאים להצלחת היירוט, כאשר ברור שכישלון במשימת היירוט יכול לגבות מחיר גבוה מאוד. הבעייתיות של תהליך זה מתחילה כבר בחישנים: לכל סוג חישן יש מגבלות של ביצועים שמתבטאות בטווח הגילוי, דיוק המדידות, קצב העברת הנתונים, התראות שווא וכו'. הבעיה נעשית יותר קשה כאשר מדובר במספר חישנים, ובמיוחד כאשר החישנים האלו מסוגים שונים – נדרש לעשות היתוך מידע המגיע מהחישנים ולעבד אותם ביחד. ישנה דילמה עבור העוקב: פחות מדי דיווחים מהחישנים יגרמו לאי דיוק בעקיבה, כאשר ריבוי הדיווחים יגרום להצפת מידע. הבעיה הולכת ומסתבכת כאשר מדובר במספר איומים: יש מספר עצמים שמאיימים על האזור המוגן ויש צורך במספר טילים כדי ליירט אותם. במקרה זה המורכבות של כל התהליך עולה, יש יותר דיווחים, יותר מטרות, יותר חישובים ועדיין ישנה דרישה לזמני תגובה מספיק מהירים מצד אחד, ושערוך המיקום באופן מספיק מדויק מצד שני.



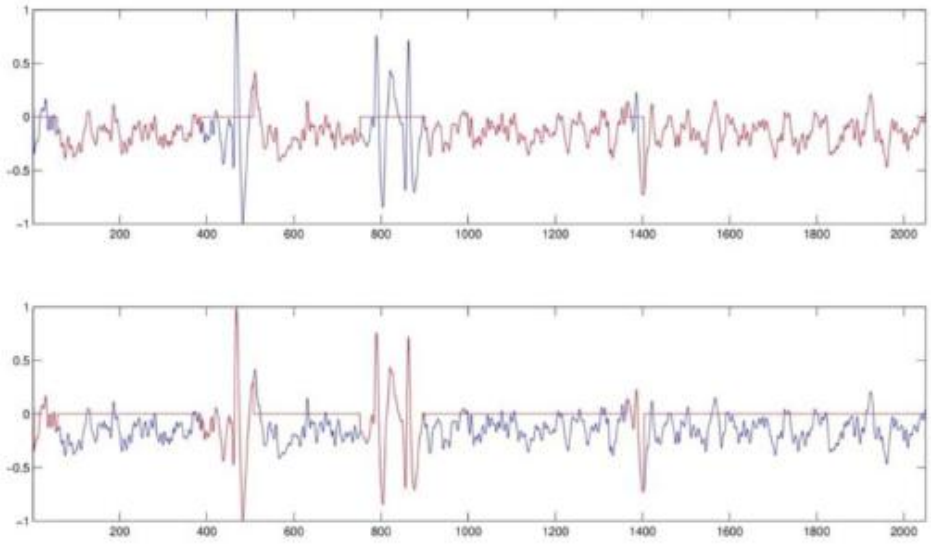
איור 1: דוגמא למערכת משולבת חישנים לוויינים לגילוי ויירוט טילים STSS.

באיור 1 שנלקח מ- http://missiledefense.wordpress.com מתוארת אילוסטרציה למערכת מסוג זה.דוגמא למערכת מהתחום האזרחי: מערכת בקרה אווירית. מטרת מערכת זו היא לתת תמונת מצב עדכנית ומדויקת של השמיים בתחום אחריות הבקרה. שמי כל העולם מחולקים לאזורי בקרה שונים. באחריות הבקר להכיר כל מטוס שנע באזור האחריות שלו, לוודא שהוא מתקדם לפי תוכניות הטיסה, לזהות חריגות ולנהל את התקלות המתעוררות. כל המטוסים אמורים לנוע באותם נתיבים, גבהים ומהירויות שהוגדרו עבורם ע"י הבקר. המערכת אמורה לספק לבקר את תמונת השמיים, כלומר את הנתונים הבאים: מיקום המטוס, מהותו (נתוני זיהוי), מהירותו וגובהו. לפי הנתונים שלעיל ניתן, מצד אחד, להסיק האם המטוס נע בהתאם לתוכנית שהוגדרה עבורו ולהתריע כאשר ישנן חריגות; ומצד שני, במקרי חירום, מכיוון שהתמונה הינה עדכנית, ניתן למצוא פתרונות אודות שינוי הנתיבים על מנת לפתור את התקלות. למשל, מטוס מסוים גילה תקלה באחד ממנועיו ונדרש לנחות באחד משדות התעופה הקרובים שלא בהתאם לתוכנית. לכן נדרש לבנות נתיב חדש עבור המטוס, וכמו כן יש לשנות את נתיבי המטוסים האחרים שעלולים להתנגש בו בעקבות השינוי. יש לעקוב אחר כל אחד מהמשתתפים ולוודא כי הם מבצעים באופן מדויק את השינויים בתוכנית ולהתריע במקרה של אי התאמה. ברור שחריגה מהתוכנית עלולה לגרום לתאונה. גם במקרה מערכת בקרה אווירית ישנם מספר חישנים שמגלים את מיקומם של העצמים שנעים בשמיים. החישנים מדווחים על הנתונים שלהם לעוקב, שבונה מסלולים עבור כל אחד מהעצמים. כמו כן, המערכת יודעת לזהות כל אחד מהמטוסים ולבדוק את מידת ההתאמה של התנועה הנמדדת והמשוערכת לזו המתוכננת. גם במקרה זה סיבוכיות המערכת עולה כתלות בכמות המטרות שיש לעקוב אחריהם. גם במקרה זה ישנה דרישה לתת שיערוך מדויק בקבועי זמן מאוד קצרים.



איור 2: דוגמא לתמונת מכ"ם במערכת הבקרה האווירית ATC.

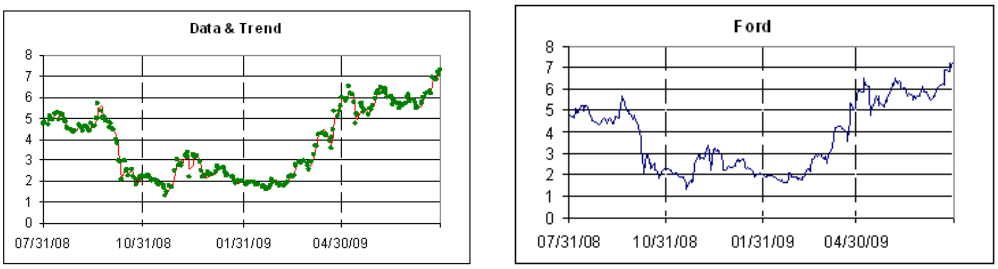
באיור 2 שנלקח מאתר http://airtrafficcontrollerexam.com מתואר מסך של מערכת בקרה האווירית. בעיית העקיבה רלוונטית לא רק עבור תחום התנועה האווירית. היא שימושית בתחומים שונים כגון הביולוגיה והרפואה [OTKTF2009]. במאמרם זה, קבוצת החוקרים מתארת את השימוש בטכנולוגיה של עוקב לשם אבחון מחלת האפילפסיה ומחלות מוח אחרות. אחת השיטות המקובלות בעולם הרפואה כיום היא בדיקת האלקטרואנצפלוגרף EEG (electroencephalography). בשיטה זו מחברים מספר חישנים לגולגולתו של הנבדק ורושמים את גלי המוח הנוצרים על ידי פעילות ספונטנית מוחית. המאפיין שמתאר את הגל הוא התדר שלו. פעילות המוח שמעניינת מבחינה קלינית מורכבת מ- 5 מקצבים (rhythm): אלפה, ביתה, גמה, דלתה וטתה. התדרים של המקצבים האלו נמצאים בתחום בין 0.1-100 Hz. שיטת הבדיקה היא לגלות קפיצות לא לינאריות או גלים חדים במיוחד. התופעות האלו נגרמות בדרך כלל ע"י פרוק מטען חשמלי סימולטני בקבוצת הנירונים שנפחה מספר מילימטרים מעוקבים. כמו במקרה של מדידת מיקומו של המטוס, בתוך מדידת גלי המוח ישנו מרכיב של רעש שנובע ממגבלות אמצעי המדידה. יש צורך להוריד רעשים אלו על מנת להתמקד בתופעות האמיתיות. במקרה זה תפקידו של העוקב הוא לתפקד כמסנן שמחליק את המדידות וכך קל יותר לזהות את הקפיצות בתדר שנובעות מהמחלה.



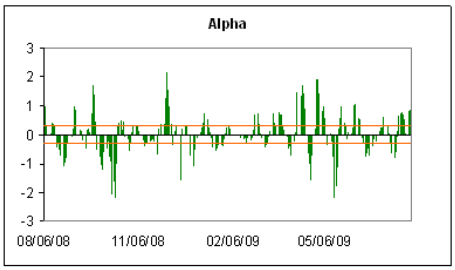
איור 3: דוגמא לאלקטרואנצפלוגרף.

איור 3 נלקח מ [OTKTF2009]. בשני התרשימים הקו הכחול מסמן את המדידות שעברו החלקה. הקו האדום בתרשים הראשון מסמן את הפעילות שנחשבת כרגילה, לעומת זאת, בתרשים השני הקו האדום מסמן את החלק המשלים למקרה הראשון: הוא מסמן את הקפיצות החריגות שעלולות להיות בעייתיות.

דוגמא נוספת שאפשר להביא לשימוש בעקיבה היא מתחום הכלכלה. Parischa [P2006] מציג מספר שימושים שניתן לעשות לעוקב של Kalman Filter הן בתחום מקרו כלכלה – כגון בניית מודל של שינוי שלטון המבוסס על מודלים מרקוביים וכן במיקרו כלכלה כאשר הוא מביא דוגמא לשערוך פרמיית סיכון במסחר שערי חליפין. כמו כן Kalman Filter הוא אמצעי לחישוב של alpha indicator [MR2009] האחרון הוא אמצעי למדידת ביצועים של השקעה כגון קרן ונחשב לאחד מחמשת המדדים העיקריים. האינדיקאטור הזה מציג את השוני בין הביצועים בפועל של הקרן יחסית לצפי המקורי.



(a) (b)



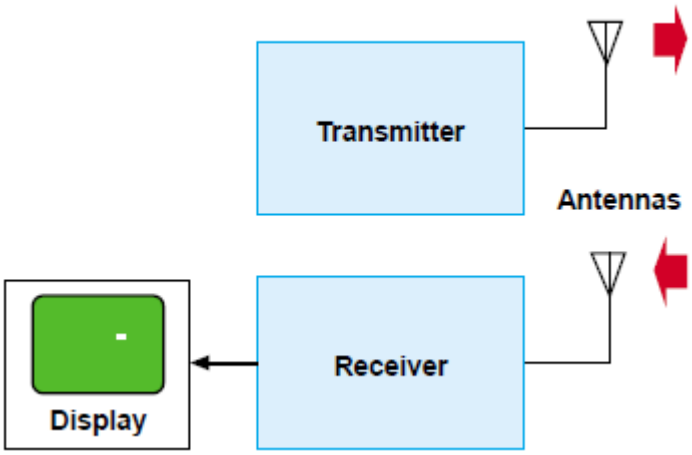
(c)

איור 4: (a) דוגמא לנתוני מסחר של חברת פורד (b) שהוחלקו ע"י Kalman Filter (c) וחישוב הפרמטר Alpha .

איור 4 נלקח מ- [MR2009]. כפי שכבר נאמר, במערכות עקיבה למטרות מוטסות מורכבות הבעיה מתחילה בחישן. מטרת החישן היא לגלות את קיום העצם ולשערך את מיקומו וכמו כן, מה שלא פחות חשוב, לדווח את שגיאת המדידה שלו. ישנם הרבה סוגים של חישנים: אלקטרומגנטיים, סונר, אופטיים וכו' [B2003]. את החישנים האלקטרומגנטיים אפשר לחלק לשני סוגים:

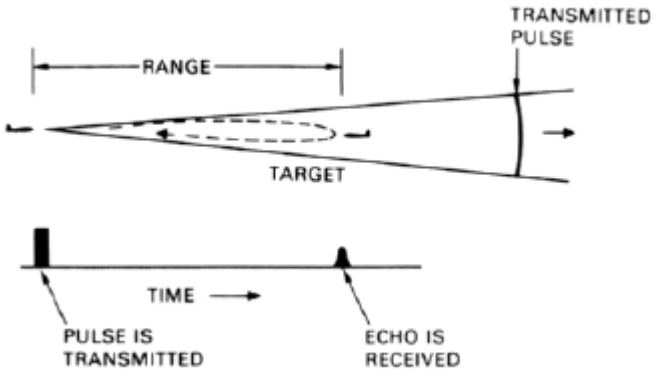
1. חישנים פעילים, כגון מכ"מ.
2. חישנים סבילים, כגון מגלה כיוון.

עיקרון פעולת המכ"מ [S1998], המגלה כיוון ומרחק (radar – range and distance), מבוסס על התכונה הפיסיקלית של עצם להחזיר את הקרינה האלקטרומגנטית שהוא מקבל. מכיוון שמהירות התפשטות הגל היא ידועה וקבועה – הרי זאת מהירות האור, ניתן להשתמש במבנה הבא:



איור 5: תאור פשטני של מכ"מ.

איור 5 נלקח מ- [S1998]. המשדר משדר את האות האלקטרומגנטי שפוגש בדרך איזשהו עצם. העצם מחזיר את האות חזרה שבסופו של דבר נקלט ע"י המקלט. האות המוחזר מעובד ומוצג בתצוגה. לדוגמא,



איור 6: גילוי המרחק בעזרת המק"מ המוטס.

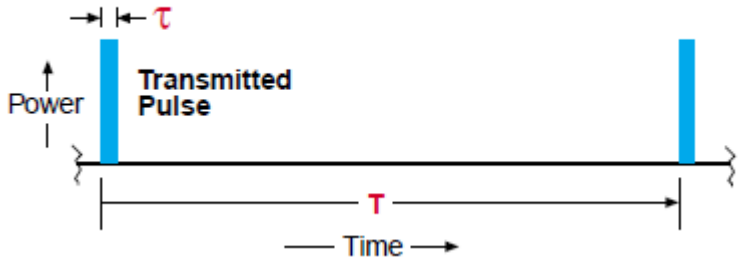
באיור 6, שנלקח מ- [S1998], בחלקו העליון, ניתן לראות בצד שמאל מטוס המצויד במכ"מ שמשדר את האות (פולס) המסומן בקו מקווקו. הפולס פוגע במטוס שממול ומוחזר ממנו ובסופו של דבר נקלט במטוס ה"מתשאל". בחלקו התחתון של הציור ניתן לראות את ה"היסטוריה" של האות: בצד שמאל נמצא הפולס המשודר וכעבור זמן מה הפולס חוזר מוחלש ומעוות. כאמור, מכיוון שמהירות האור היא קבועה, ניתן לחשב את הטווח בקלות. נניח, האות חזר לאחר 10 מיקרו שניות, אזי המרחק בין המכ"מ לעצם מחושב בצורה הבאה:

$$R = \frac{1}{2}\ \text{(Round-Trip Time)} \times \text{(Speed of Light)}$$

$$= \frac{1}{2} \times \frac{10}{1{,}000{,}000}\ \text{s} \times 300{,}000{,}000\ \text{m/s}$$

$$= 1.5\ \text{km}$$

יש לשים לב למקדם של החצי: האות עושה את הדרך פעמיים: מהמכ"מ עד לעצם ובחזרה. בדרך כלל האות המשודר במכ"מ הוא פולס.

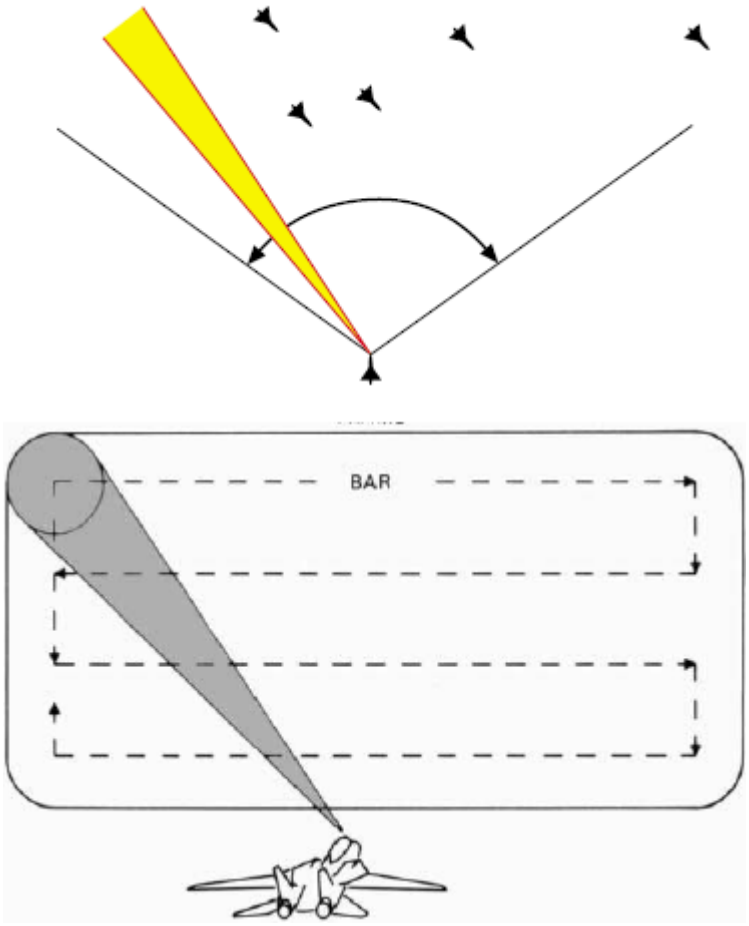


איור 7: תאור סכמטי של פולס המכ"מ.

איור 7 נלקח מ- [S1998]. הרעיון בשימוש בפולס הוא לשלוח את האות ולחכות לתשובה, ורק אחר כך לשלוח את הפולס הבא. זמן T צריך להיות ארוך מספיק בהתאם למטרות המכ"מ. אם זמן t המסמן את ה- round trip של הפולס מקיים את t>T, אזי המכ"מ יכול להתבלבל ולחשוב שבעצם זמן ה- round trip הוא t-T כיוון שאולי הוא חזר בעקבות שידור הפולס השני ואז

המרחק המחושב יהיה שגוי. מצב מעין זה נקרא ambiguity. בתורת המכ"מ יש מגוון פתרונות שמוצעים לבעיה זו, אבל כאמור, צריך להתאים את הפרמטרים של השידור לאופי הזירה שמכ"מ אמור להתמודד איתה.

במציאות המכ"ם סורק את הזירה ובצורה זו מקבל את תמונת המצב:



איור 8: תאור סריקה של מכ"מ.

איור 8 נלקח מ- [S1998]. בחלק העליון של האיור מוצג מטוס הסורק את הזירה מולו ובחלק התחתון מוצגת "תוכנית" הסריקה שהיא דו- ממדית (שמאל-ימין: ציר ה-x, מעלה-מטה: ציר ה-y). חשוב להבין שככל שהאלומה (הסקטור במרחב שמואר ע"י המכ"מ) צרה יותר, התוצאות יוצאות מדויקות יותר. הסיבה לכך היא שהאנרגיה שמתבזבזת במרחב קטנה יותר ולכן האות שיוחזר יהיה חזק יותר וגם כי ההסתברות שיהיו שני עצמים בתוך האלומה קטנה. מצד שני, אלומה צרה מאריכה משמעותית את זמן הסריקה ואז כל האזור נמצא באזור ה"עיוור" לזמן רב יותר.

איכות חישוב הטווח תלויה בהרבה פרמטרים נוספים:

- עוצמת השידור
- גודל האנטנה (משדרת/קולטת)
- התכונות הפיסיקליות של העצם המחזיר – "טיב" ההחזרה

- זמן החשיפה של העצם לאות המכ״מ ואורך מחזור הסריקה
- אורך הגל שבו משודר האות
- רעש רקע

ישנם שימושים רבים למכ״מים:

- חיזוי מזג האוויר
- ניווט
- מיפוי הזירה (בקרה אווירית, בקרה ימית)
- חיפוש ועקיבה
- הכוונת טילים
- גילוי איומים (טילים)
- זיהוי מטרות (זיהוי עמית – טורף)
- מיפוי שטח ברזולוציה גבוהה מאוד (מכ״מים מסוג SAR)

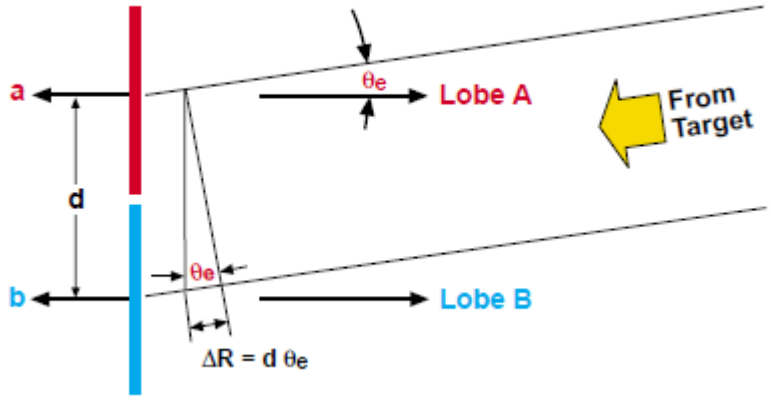
היתרון של השיטה האקטיבית הוא ברור: בהינתן מיקום המכ״מ והאזימוט של האנטנה, ניתן בקלות למפות את הזירה.
ישנם גם חסרונות לחישן האקטיבי:

1. המחיר והמורכבות. מכיוון שמכ״מ משדר בתדר גבוה מאוד ובצורה מאוד מדויקת – עלותו גבוהה. כמו כן, נדרשים אמצעים נוספים כגון מקור להספקת כוח משמעותי, מערכות קירור וכו׳.
2. חתימת פעילות גבוהה. מכיוון שהמכ״מ משדר – הוא נקלט ע״י כולם, וכך החשאיות נאבדת. ניתן ללמוד את שגרת הפעילות, להבין את חסרונות הפריסה ואפילו קל להשמידו במידת הצורך.

חסרונות אלו יכולים להיות קריטיים כאשר מתכנים פתרון של עקיבה. ישנה דרך חילופית והיא שימוש בחישנים פסיביים. חישנים אלו לא משדרים אלא רק קולטים את האותות המשודרים.
המקורות של השידור יכולים להיות שונים ומגוונים:

- שידורי מכ״מ
- שידורי תקשורת
- שידורי טלמטריה (שיטה למדידה מרחוק ודיווח של מדידות) וכדומה

ברור, שבמקרה של שימוש בחישן פסיבי שיטת האיכון היא שונה: בדרך כלל לא ידוע מתי שודר שידור כלשהו ולכן אי אפשר להשתמש בנוסחא לחישוב הטווח. אחת השיטות המקובלות לאיכון פסיבי של המטרות הוא במגלה הכיוון.



איור 9: תאור המערך לגילוי הכיוון.

איור 9 נלקח מ- [S1998]. בשיטה זו לחישן יש שתי אנטנות a ו- b שנמצאות במרחק d אחת מהשנייה. האות נקלט באופן שונה בשתי האנטנות כפונקציה של θ_e שהוא הכיוון שבין המטרה למערך האנטנות. בקלות ניתן לחשב את האזימוט של המטרה מהחישן כאשר מקובל לחשב את האזימוט יחסית לצפון. בהמשך העבודה תתואר הדרך שבה ניתן לשערך את מיקום המטרה בעזרת מספר חישנים.

התוצאה שמתקבלת בחישן אינה תוצאה מדויקת. במקרה של מכ"מ ישנן טעויות במדידת זמני שידור וקליטה, התדר אינו מדויק, יש טעויות במדידת עוצמת האות, מהירות האור לא באמת קבועה כי היא תלויה בתווך שבו הוא עובר וכו'. גם בגילוי כיוון יש תלות בטעויות המדידה: מרחק בין האנטנות, מדידת הפרשי הפאזה החשמלית של אות הנקלט בערוצים שונים וכו'. כתוצאה מכך, לא ניתן למצוא בצורה מדויקת בהחלט את מיקום המטרה אלא רק לתאר התפלגות כלשהי של המיקום. יחד עם זאת, לא קיים צורך אמיתי בדיוק מוחלט. במקרה של עקיבה אחרי טיל רצוי לקבל דיוק של מטרים ספורים, במקרה של בקרה אווירית גם דיוק של עשרות מטרים הוא מקובל ובמקרה של מכ"מ מזג האוויר אפילו ניתן להסתפק בדיוק קטן מזה.

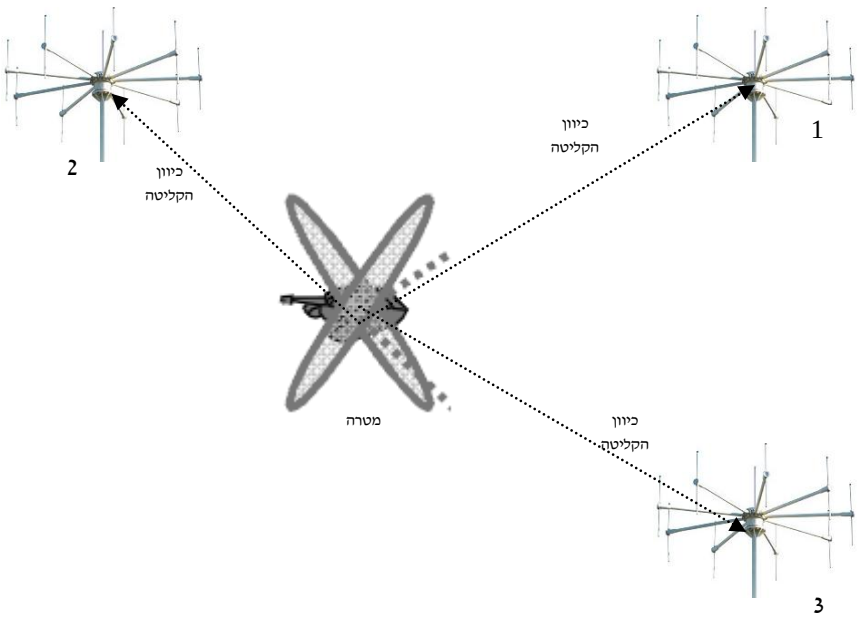
3 סקירת ספרות

ישנם מקורות רבים שעוסקים בנושאים של איכון, עקיבה וגם מבני נתונים מרחביים. הנושאים האלו זוכים להתייחסות בתחומים כגון בקרה, גאומטריה חישובית וכו׳.
נושא האיכון התחיל להיחקר כבר באמצע המאה הקודמת בשל חשיבותו לתחום הביטחוני. בשנת 1947, [S1947] פרסם עבודה שבה תוארו שיטות בסיסיות לביצוע איכון על המטרה בעזרת חישנים פסיביים. [A1958] מרחיב ומסביר אותו ביתר פירוט. בהמשך [T1984] עושה סקירה רחבה של שיטות איכון פסיביות הקיימות בשלב זה תוך כדי שהוא מציע שיפורים וכלים מתמטיים חדשים כדי להתגבר על הבעיות שעלו באלגוריתם של [S1947]. בין השאר הוא מתמקד בשימוש בשערוך MLE על מנת לשערך את מיקום המטרה. נושא שערוך אליפסת אי-ודאות בהינתן התפלגות נורמלית דו-ממדית עולה במאמרים רבים: [T1984], [R2004] ואחרים.
[G1995], בעבודת הדוקטורט שלו מתייחס, בין השאר, לשיטות שהוזכרו ב-[T1984], אבל עושה פיתוח נוסחאות מחדש תוך כדי התייחסות לעקמומיות כדור הארץ.
נושא העקיבה גם הוא זוכה להתייחסות ולמחקרים רבים. בשנת 1960, [K1960] פרסם מאמר שבו הוא מתאר את השימוש ב-Kalman Filter לצורך בקרה על מערכות דינמיות. כאמור, נושא זה נמצא בשימוש בהרבה תחומים שונים כגון רפואה וביולוגיה [OTKTF2009], כלכלה [P2006] וכמובן גם בנושא העקיבה [MBCC1999], [EM2001], [G1995], [R2004] , [SBC1999] וכו׳.
במהלך העבודה עם ה-Kalman Filter עלו מגבלות השיטה הזו ולכן עלו רעיונות לשיפור כגון Adaptive Kalman Filter [HCCL2003], Extended Kalman Filter [G1995], [R2004] וגם Unscented Kalman Filter [BM2007]. מצד שני, חלה גם התפתחות בנושא העקיבה אחרי מספר מטרות. אחת העבודות החשובות בנושא נעשתה ע״י [B2003] שבה הוא מתאר את המושגים עיקריים של עקיבה לאחר מספר מטרות כגון סריקה, מטרה, מסלול וכו׳. כמו כן פורסמו עבודות של [R1979], [SS1971], [KNS2004], [B1995] שמתארות שיטות שונות של עקיבה לאחר מטרות מרובות .
נושא התאור הפורמלי לבעיית החיפוש נעשה ע״י [C1990] ו-[C1995]. בהמשך נעשית סקירה רחבה של נושא החיפוש במבנים גאומטריים ע״י [AE1997] ו- [GG1997]. הספר [BKOS2000] מתאר מספר נושאים של גאומטריה חישובית ובין השאר מתאר את האלגוריתמים לחיפוש בתחומים, סגמנטים, עצי רביעים וכו׳. [RG2003] מתאר את עצי B/B+ ושימושם באינדוקס מסדי נתונים. [G1984] מתאר את עצי R/R+ וכו׳.
חשוב לציין מספר ספרים יסודיים: [BD2001] בנושא של סטטיסטיקה, [CLR1990] בנושאים של מבני נתונים ואלגוריתמים ו-[BKOS2000] בנושא של גאומטריה חישובית.

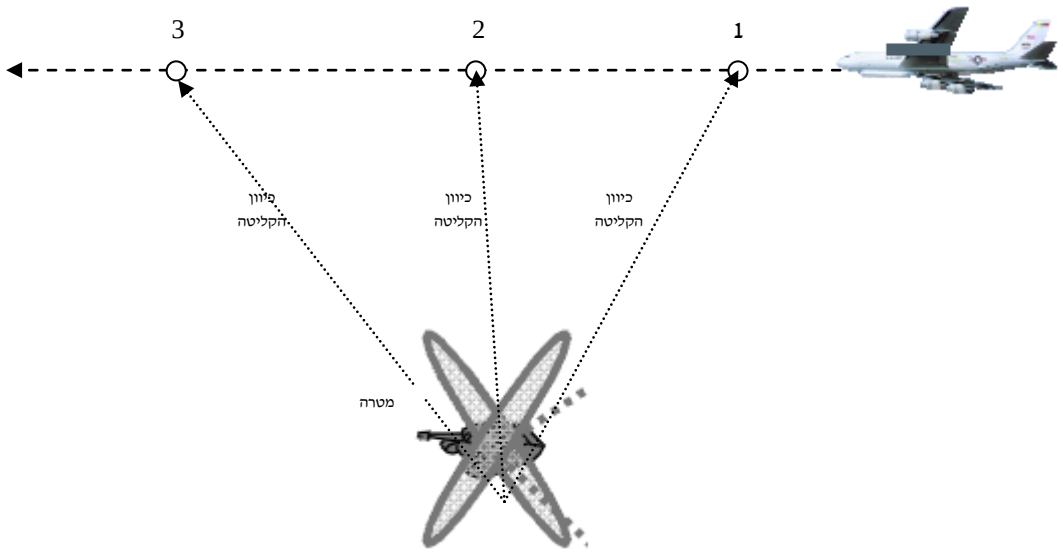
4 איכון המטרה

4.1 הגדרת המודל

השלב הראשון בעקיבה אחר מטרה מוטסת הוא איכונה. החישנים של מערכת העקיבה מגלים את המטרה ומודדים את הנתונים שתלויים במיקומה. כפי שנאמר בפרק המבוא, ישנם הרבה גורמים לאי-דיוק במדידה. בעצם כל אחד מהחישנים הוא תהליך סטוכסטי או תהליך מקרי בדיד שמוגדר כסדרת משתנים מקריים $\{X_n\}_{n\in T}$ או $\{X_n; n\in T\}$ על מרחב ההסתברות (Ω, B, P), כאשר T היא קבוצת האינדקסים [G2009]. במלים אחרות, כל מדידה של כל אחד מהחישנים היא בעצם משתנה מקרי שמתפלג בהסתברות מסוימת. לכן מיקום המטרה יכול להתקבל רק כשיערוך סטטיסטי. ההתפלגות של המשתנה המקרי X_n תלויה במודל שיוגדר. כמובן, ככל שהמודל יהיה טוב יותר, כך גם התוצאה תהיה מדויקת יותר. בשנת 1984, D.J.Torrieri [T1984] פרסם מאמר שמגדיר את מודל המדידה וגם מסביר את שערוך מיקום המטרה עבור מערכות פסיביות. במאמר זה מוצגים שני מודלים של מערכות איכון: האחד מבוסס על מערך חישנים שמגלים את כיוון המטרה יחסית לחישן והשני מבוסס על החישנים שמודדים את זמן קליטת השידור. בסוף המאמר מוצגת דרך לשערך את וקטור המהירות של המטרה בעזרת שימוש בתופעת Doppler.
בפרק זה יוסבר מודל האיכון שמבוסס על חישנים שמגלים כיוון. מודל זה יכול לעבוד בשני מקרים שקולים.
מקרה א'



מקרה ב':



איור 10: דוגמא לקליטת המטרה.

במקרה א' המטרה הנייחת נקלטת ע"י שלושה חישנים נייחים. בכל אחד מהחישנים המטרה נקלטת בכיוון אחר. במקרה ב' ישנו אומנם חישן אחד, אבל הוא נייד ולכן בכל אחת משלוש נקודות הזמן הוא נמצא במקומות שונים ושוב המטרה נקלטת מכיוונים שונים. ברור, שמהירות החישן הנייד צריכה להיות גדולה משמעותית ממהירות המטרה, אחרת מיקום המטרה לא יהיה מדויק.

תהליך האיכון יוסבר על הדוגמא של מקרה א'.

בעקבות קליטת השידור מתקבלות מדידות מהחישנים:

$$r_i = f_i(x) + noise_i, \quad i = 1,2,..., N \tag{1}$$

כאשר N הוא מספר החישנים, i הוא האינדקס של החישן, $f_i(x)$ היא הפונקציה של המדידה שהייתה מתקבלת אם לא היו שגיאות במדידה, $noise_i$ היא פונקצית השגיאה של המדידה מהחישן i ובסוף r_i היא המדידה שמדווחת ע"י החישן.

אפשר לכתוב את N המשוואות בצורה וקטורית:

$$r = f(x) + noise \tag{2}$$

כאשר $noise$ הוא וקטור מקרי עם מטריצת שונות (covariance matrix) משותפת חיובית סימטרית

$$N = E\left[(noise - E[noise])(noise - E[noise])^T\right] \tag{3}$$

המשמעות של משתנה x שלעיל הוא צירוף של המיקומים של המטרה והחישן שאינה ידועה מראש, אבל אינה משתנה בצורה מקרית. בהינתן זירה מסוימת מיקומה של המטרה ביחס לחישן הוא קבוע אבל לא ידוע. נדרש לגלות את מיקומה היחסי של המטרה. ולכן, כפי שמקובל, מניחים

ש-x הוא משתנה שאינו ידוע אך אינו מקרי. לעומת זאת, ה- noise הוא מקרי ומתפלג בצורה גאוסיאנית, לכן אפשר לכתוב את צפיפות הסתברות של המדידה בצורה הבאה:

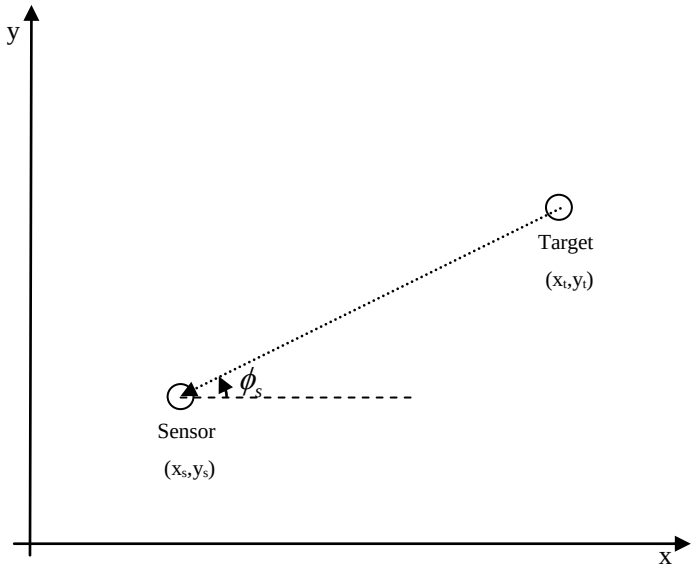
$$p(r \mid x) = \frac{1}{(2\pi)^{N/2}|N|^{1/2}} \exp\left\{-(1/2)[r - f(x)]^T N^{-1}[r - f(x)]\right\} \qquad (4)$$

להלן מספר הסברים:

המטריצה N הפיכה בשל היותה מוגדרת כחיובית וסימטרית.

$r - f(x)$ הוא בעצם n - שהיא טעות המדידה.

נגדיר עכשיו את הפונקציה *f(x)*, שהיא פונקציית מדידה של המטרה ע"י החישן במקרה האופטימלי, נטול שגיאות. אם החישן מודד את זוית ההגעה של האות אזי המשואה (5) מתארת את *f(x)*.על מנת למנוע בלבול, נשתמש באות R להיות הארגומנט של הפונקציה במקום x.



איור 11: הגדרת פונקציית כיוון בין המטרה לחישן.

לכן,

$$f(R) = \tan^{-1}\frac{y_t - y_s}{x_t - x_s} \qquad (5)$$

ואם נוסיף את פונקצית השגיאה נקבל

$$\phi_i = \tan^{-1}\frac{y_t - y_i}{x_t - x_i} + n_i \qquad (6)$$

כאשר ϕ_i הם דיווחי הכיוון שמדווחים ע"י החישנים. בעקבות דיווחי מדידות של הקליטה עשויה להתקבל התמונה הבאה:

איור 12: מדידת הכיוונים בחישנים.

המשולשים השחורים מסמנים את החישנים, הריבוע האדום הוא המטרה והקווים הכחולים הם הקוונים שנקלטו ע״י החישנים. כפי שניתן לראות, המטרה אינה נחתכת בצורה מדויקת ע״י הקוונים (הקרן שמקורה בחישן וזויתה נמדדת על ידי החישן) ולכן הפתרון שהוא מציאה אנליטית של נקודת החיתוך של הקוונים לא ישים ואפילו לא קיים. דרך ההתמודדות שמציע [T1984] במאמרו היא התמודדות סטטיסטית בעזרת שימוש במשערך MSE (Maximum Likelihood Estimator) [BD2001].

4.2 שיערוך המיקום

השיטה מתוארת ב- [T1984]. נחזור לנוסחא (4). היא מתארת את צפיפות הסתברות קבלת מדידה בהינתן מיקומים קבועים וידועים של החישנים ומיקום קבוע אך לא ידוע של המטרה. למצוא את $MSE(p(r|x))$ זה בעצם למצוא את הערך של x כזה שימקסם את ה-p. או במילים אחרות, מכיוון שהפרמטר הלא ידוע היחיד הוא מיקום המטרה, שערוך MSE ישערך מיקום כך שההסתברות לקבל משם את המדידות שנמדדו תהיה מקסימלית. אחת הטכניקות המקובלות בחישוב MSE על התפלגויות גאוסאניות זה להפעיל פונקצית log על (4)

$$Q(x) = [r - f(x)]^T N^{-1} [r - f(x)] \tag{7}$$

ולחפש את x כך ש-Q(x) יהיה מינימלי. השינוי מחיפוש מקסימום למינימום נובע מהסימן מינוס שנמצא בתוך המעריך.

פתרון בעיית מציאת נקודת הקיצון של הפונקציה הוא לגזור אותה ולמצוא את הנקודה שבה היא שווה ל-0. מכיוון שמדובר בפונקציה עם מספר משתנים ככמות החישנים, נדרש למצוא את הערך של x שמביא את הגרדיאנט של G(x) לערך 0.

$$\nabla_x Q(x) = \left[\frac{\partial Q}{\partial x_1}, \frac{\partial Q}{\partial x_2} \ldots \frac{\partial Q}{\partial x_N} \right]^T \tag{8}$$

ובעצם נדרש לפתור N משוואות מסוג $\frac{\partial Q}{\partial x_i} = 0$ עם N נעלמים $\{x_1, x_2, \ldots, x_N\}$

מכיוון שלפי (6) הפונקציה $f(x)$ אינה פונקציה לינארית, אז מציאת הפתרון של מערכת משוואות לא לינאריות ייעשה בשיטה הנומרית של Newton-Gauss

$$f(x) \cong f(x_0) + G(x - x_0) \tag{9}$$

ו- G היא מטריצת נגזרות חלקיות של פונקציות $f(x)$ לפי כל אחד מהמשתנים

$$G = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{x=x_0} & \cdots & \cdots & \left.\frac{\partial f_1}{\partial x_N}\right|_{x=x_0} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \left.\frac{\partial f_N}{\partial x_1}\right|_{x=x_0} & \cdots & \cdots & \left.\frac{\partial f_N}{\partial x_N}\right|_{x=x_0} \end{bmatrix} \tag{10}$$

אם נגדיר את $\hat{x}$ בתור המשערך המביא את $Q(x)$ במשוואה (7) לערכו המינימלי, ונגדיר משתנה חדש r_1 כך ש-

$$r_1 = r - f(x_0) + Gx_0 \tag{11}$$

אז, אם נציב את (9) בתוך (7) ונגזור, נקבל

$$\nabla_x Q(x)\Big|_{x=\hat{x}} = 2G^T N^{-1} G\hat{x} - 2G^T N^{-1} r_1 = 0 \tag{12}$$

אם $G^T N^{-1} G$ אינה סינגולרית אז היא הפיכה ולכן פתרון המשוואה הזו יהיה

$$\begin{aligned} &\hat{x} = \left(G^T N^{-1} G\right)^{-1} G^T N^{-1} r_1 = \\ &x_0 + \left(G^T N^{-1} G\right)^{-1} G^T N^{-1} [r - f(x_0)] = \\ &x + \left(G^T N^{-1} G\right)^{-1} G^T N^{-1} [f(x) - f(x_0) - G(x - x_0) + n] \end{aligned} \tag{13}$$

שוב, כדי לעשות סדר במשתנים: x מייצג את המיקום האמיתי של המטרה, x_0 - ניחוש ראשוני של נקודה כלשהי שנמצאת בסביבת המטרה, $\hat{x}$ - הפתרון שניתן בסוף האלגוריתם מטריצת השונות המשותפת של $\hat{x}$ תחושב לפי (13) בצורה הבאה:

$$
\begin{aligned}
&P = Cov(\hat{x}) = E[\hat{x}\hat{x}^T] = \\
&\left(G^T N^{-1} G\right)^{-1} G^T N^{-1} r_1 r_1^T N^{-1} G \left(G^T N^{-1} G\right)^{-1} = \\
&\left(G^T N^{-1} G\right)^{-1} G^T N^{-1} N N^{-1} G \left(G^T N^{-1} G\right)^{-1} = \\
&\left(G^T N^{-1} G\right)^{-1} G^T N^{-1} G \left(G^T N^{-1} G\right)^{-1} = \left(G^T N^{-1} G\right)^{-1}
\end{aligned}
\tag{14}
$$

נחזור ממקרה הכללי למקרה הפרטי כפי שתואר ב-(6).

כמו במקרה הכללי, נגדיר נקודה כלשהי בסביבת המקום הצפוי $R_0(x_0, y_0)$. מחבר המאמר מציע למצוא את מרכז המצולע של נקודות חיתוך בין הכוונים.

אם נחשב את האזימוט ביו נקודת הניחוש R_0 לכל אחד מהחישנים נקבל

$$\sin\phi_{0i} = \frac{y_0 - y_i}{D_i};\ \cos\phi_{0i} = \frac{x_0 - x_i}{D_i};\ i = 1,2,...,N \tag{15}$$

כאשר

$$D_i = \left[(x_0 - x_i)^2 + (y_0 - y_i)^2\right]^{1/2};\ i = 1,2,...,N \tag{16}$$

כדי לחשב את מטריצת G שהיא לפי (10) מטריצת הנגזרות החלקיות של הקוונים לפי x ו- y המרחביים בנקודת הניחוש. כידוע נגזרת של arctan היא

$$\frac{d(\arctan(\theta))}{d(\theta)} = \frac{1}{1+\theta^2} \tag{17}$$

לכן,

$$\frac{\partial(\arctan(\frac{y-y_i}{x-x_i}))}{\partial(y)} = \frac{1}{1+(\frac{y-y_i}{x-x_i})^2}\frac{1}{x-x_i} = \frac{x-x_i}{D_i^2} = \frac{\cos\phi_i}{D_i} \tag{18}$$

$$\frac{\partial(\arctan(\frac{y-y_i}{x-x_i}))}{\partial(x)} = -\frac{1}{1+(\frac{y-y_i}{x-x_i})^2}\frac{(y-y_i)}{(x-x_i)^2} = -\frac{y-y_i}{D_i^2} = \frac{\sin\phi_i}{D_i} \tag{19}$$

ואז מקבלים

$$G = \begin{bmatrix} -(\sin\phi_{01})/D_{01} & -(\cos\phi_{01})/D_{01} \\ \cdot & \cdot \\ \cdot & \cdot \\ -(\sin\phi_{0N})/D_{0N} & -(\sin\phi_{0N})/D_{0N} \end{bmatrix} \tag{20}$$

כדי לחשב את המטריצה N נשתמש בדיווחי דיוק של החישן $\sigma^2_{\phi i}$

$$N = \begin{bmatrix} \sigma^2_{\phi 1} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\phi N} \end{bmatrix} \tag{21}$$

אם נגדיר את משתני העזר הבאים:

$$\mu = \sum_{i=1}^{N} \frac{\cos^2 \phi_{0i}}{D_{0i}^2 \sigma_{0i}^2} \tag{22}$$

$$\lambda = \sum_{i=1}^{N} \frac{\sin^2 \phi_{0i}}{D_{0i}^2 \sigma_{0i}^2} \tag{23}$$

$$\upsilon = \sum_{i=1}^{N} \frac{\cos \phi_{0i} \sin \phi_{0i}}{D_{0i}^2 \sigma_{0i}^2} \tag{24}$$

נשתמש ב-(13) ונקבל

$$\hat{x} = x_0 + \frac{1}{\mu\lambda - \upsilon^2} \sum_{i=1}^{N} \phi_{ri} \frac{(\upsilon \cos \phi_{0i} - \mu \sin \phi_{0i})}{D_{0i} \sigma_{\phi i}^2} \tag{25}$$

$$\hat{y} = y_0 + \frac{1}{\mu\lambda - \upsilon^2} \sum_{i=1}^{N} \phi_{ri} \frac{(\lambda \cos \phi_{0i} - \upsilon \sin \phi_{0i})}{D_{0i} \sigma_{\phi i}^2} \tag{26}$$

התוצאה המתקבלת ($\hat{x}$, $\hat{y}$) היא תוצאת שיערוך האיכון שמתאימה ביותר לכלל המדידות שנעשו ע"י החישנים.

אומנם, התוצאה הזו אינה שלמה. נשאלת עדיין השאלה מהו הדיוק שלה. ברור, שתוצאה אינה מדויקת כי היא מבוססת על מדידות שכבר גילמו את השגיאה של החישנים. מה שנדרש כעת, זה להביא את מדד הדיוק של השערוך.

מחבר המאמר מקדיש פרק שלם לדיוק המשערך.

כבר הגדרנו ש- r הוא וקטור שמתפלג בצורה גאוסיאנית. לכן, לפי (13), $\hat{x}$ הוא טרנספורמציה לינארית על r ולכן גם הוא מתפלג גאוסיאנית . פונקצית הצפיפות שלה תוגדר בצורה הבאה:

$$f_{\hat{x}}(\xi) = \frac{1}{2\pi^{N/2} |P|^{1/2}} \exp\left[-\frac{1}{2} (\xi - m)^T P^{-1} (\xi - m) \right] \tag{27}$$

כאשר לפי (14),

$$m = E[\hat{x}], \quad P = E\left[(\hat{x} - m)(\hat{x} - m)^T \right] = G^T N^{-1} G = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \tag{28}$$

4.3 שיערוך של אליפסת אי-ודאות

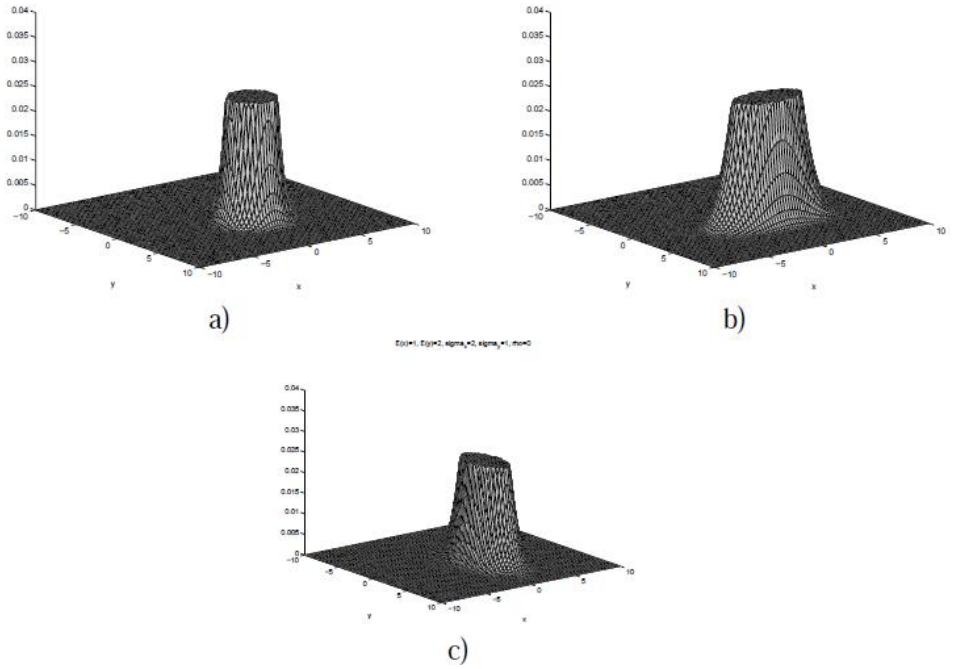
לפי [R2004], בהתפלגות גאוסיאנית דו-ממדית אוסף הנקודות שההסתברות שלהן שווה לערך K נתון היא אליפסה. במילים האחרות, אוסף הנקודות במישור שבו יכולה להימצא המטרה בהסתברות K היא אליפסה שמרכזה הוא המשערך $\hat{x}$. הסיבה לכך היא שאם ניקח פונקצית צפיפות ונפעיל עליה log נקבל

$$(\xi - m)^T P^{-1} (\xi - m) = k \qquad (29)$$

וזאת נוסחת האליפסה. במקרה הפשוט, אם מטריצה P היא אלכסונית, אז ניתן לכתוב מחדש את (29) בצורה הבאה:

$$\frac{(\xi_x - m_x)^2}{\sigma_x^2} + \frac{(\xi_y - m_y)^2}{\sigma_y^2} = k \qquad (30)$$

במקרה שמטריצה P אינה אלכסונית, כלומר יש קורלציה בין המשתנים ואז צירי האליפסה לא יהיו מקבילים לרשת הצירים.



איור 13: דוגמא לאוסף הנקודות שההסתברות שלהן קטנה מ-K.

איור 13 נלקח מ- [R2004].

בדוגמא זו יש שלושה מקרים; בשלושתם $m_x=1,\ m_y=2$.

a) $\sigma_x=1,\ \sigma_y=1$

b) $\sigma_x=2,\ \sigma_y=1$

c) $\sigma_x=1,\ \sigma_y=2$

חישוב הערכים העצמיים של P נותן

$$\lambda_1 = \frac{1}{2}\left[\sigma_x^2 + \sigma_y^2 + \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2}\right] \quad (31)$$

$$\lambda_2 = \frac{1}{2}\left[\sigma_x^2 + \sigma_y^2 - \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2}\right] \quad (32)$$

בגלל תכונותיה של P, שהיא חיובית וסימטרית, הוקטורים העצמיים $\{v_1, v_2\}$ שלה אורתוגונליים ולכן ניתן לעשות טרנספורמציה

$$P = TDT^{-1} \quad (33)$$

כך ש- $T = [v_1, v_2], D = Diag(\lambda_1, \lambda_2)$

אם נציב את (33) בתוך (29) נקבל

$$(\xi - m)^T TD^{-1}T^{-1}(\xi - m) = k \quad (34)$$

אם נסמן את

$$w(w_1, w_2) = T^{-1}(\xi - m) \quad (35)$$

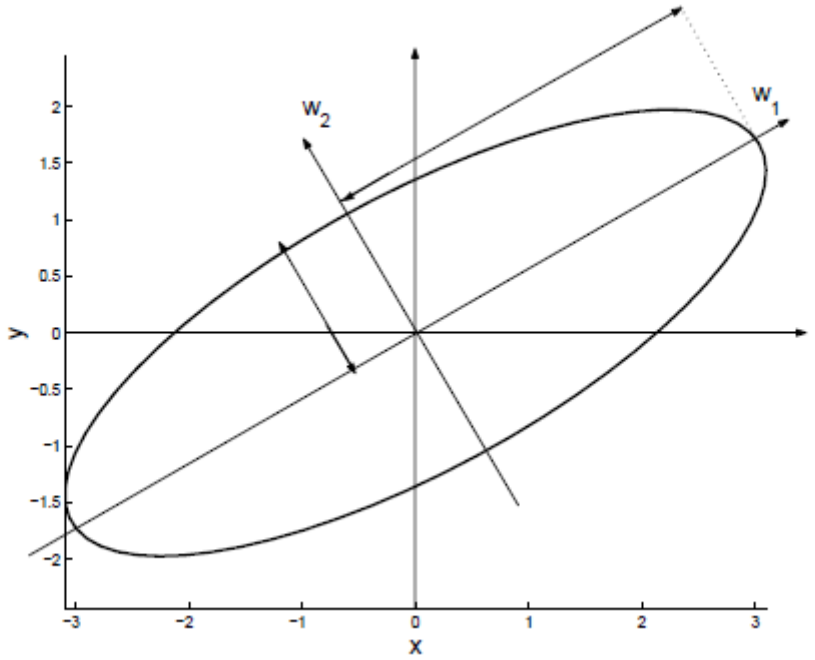
ומכיוון שבגלל האורתוגונליות $T^{-1} = T^T$

אפשר לכתוב את (29) בצורה הבאה:

$$w^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}^{-1} w = k \quad (36)$$

כלומר, בעזרת מעבר בסיס של וקטורים עצמיים, קיבלנו אליפסה עם ראשית הצירים ב- (w_1, w_2). טרנספורמציה זו היא בעצם סיבוב, לכן זווית הסיבוב היא

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right) \quad (37)$$



איור 14: דוגמא לשינוי ראשית הצירים.

איור 14 נלקח מ- [R2004]

לכן, אם ההסתברות הנדרשת היא P_e , אז

$$k = -2\ln(1-P_e) \tag{38}$$

ואז אורכי צירים של האליפסה הם

$$\begin{aligned} SemiMajor\ axis &= \sqrt{k\lambda_1} \\ SemiMinor\ axis &= \sqrt{k\lambda_2} \end{aligned} \tag{39}$$

נותר כעת לחשב את ערכי אליפסת האיכון שאת מרכזו מצאנו ב- (25) וב- (26) . נציב את (20),(21) בתוך (14). נקבל

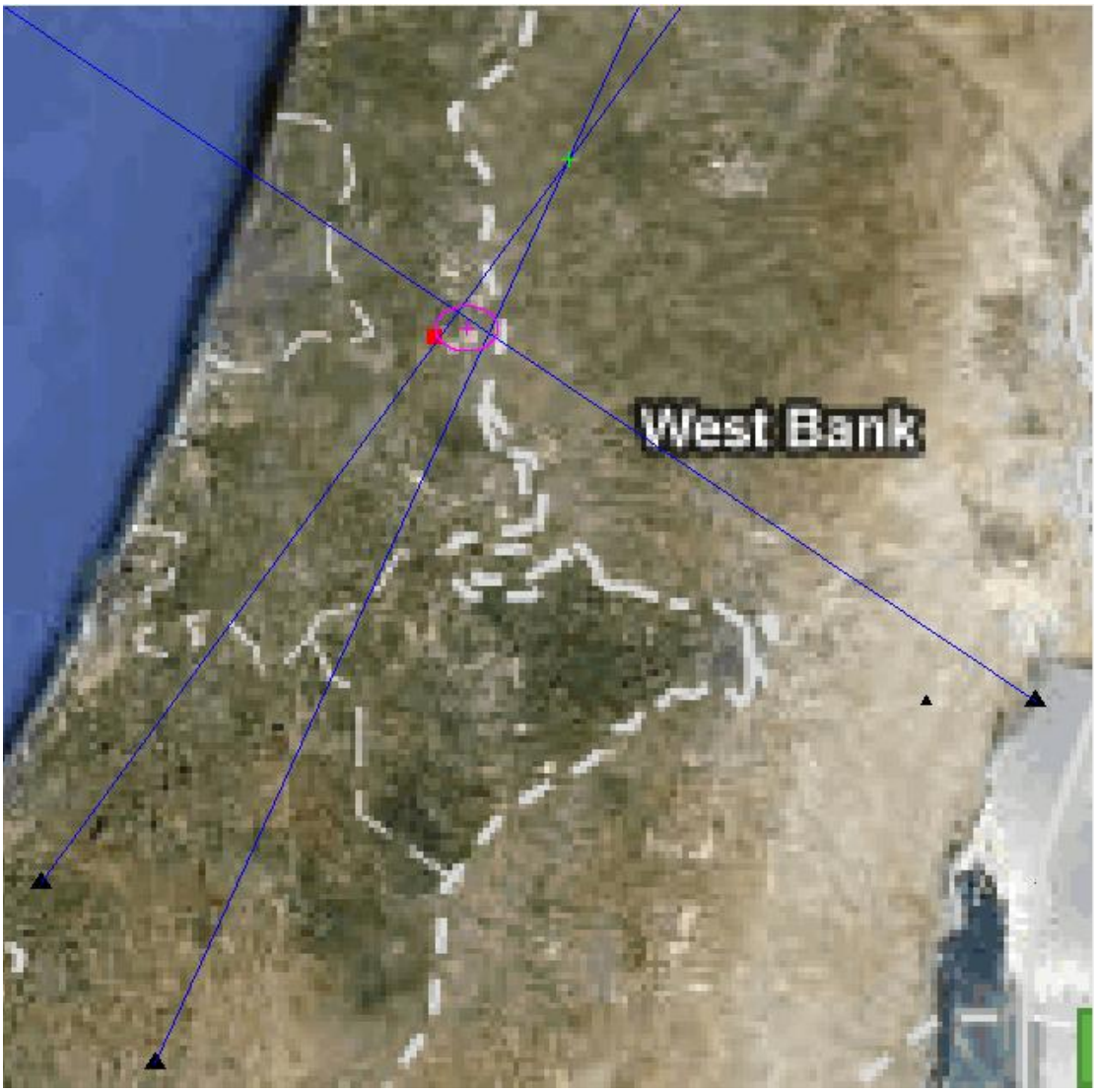
$$\sigma_x^2 = E\left[(\hat{x}-x)^2\right] = \frac{\mu}{\mu\lambda - \upsilon^2} \tag{40}$$

$$\sigma_y^2 = E\left[(\hat{y}-y)^2\right] = \frac{\lambda}{\mu\lambda - \upsilon^2} \tag{41}$$

$$\sigma_{xy} = E\left[(\hat{x}-x)(\hat{y}-y)\right] = \frac{\upsilon}{\mu\lambda - \upsilon^2} \tag{42}$$


Tebnine
El Fokhar
نافورة
Naqoura
عين ايل
Ain Ebel
119
شهبا
Shahba
السويداء
As Suwayda
الرمثا
Ar Ramtha
110
اربد
Irbid
109
Mafraq
Al Mafraq
نعيمة
Neaime
عجلون
Ajloun
جرش
Jarash
10
65
الثغرة
Al Thughra
الزرقا
Al Zarqa
بلقا
Balqa
عمان
Amman
West Bank
نوار
Naour
40
الجيزا
Al Jiza
الزميلة
Al Zumayla
65
Gaza Strip
سواقة الغربية
Suwaqa Al Gharbiya
المزرعة
Al Mazraa
كرك
Karak
القطرانة
Al Qatrana
العريش
El Arish
صافي
Safi
15
معدن
Maadan
ישראל
Israel
أبو أوجيلة
Abu-Aweigila
فيفا
Feifa
الحسا
Al Hasa
قصيمة
Quseima
5
جرف الدراويش
Jurf Al Darawish
بير حسنة
Bir Hassana
Google
Map data ©2010 Google, LeadDog Consulting, ORION-ME, Imagery ©2010 TerraMetrics

(a)



(b)

איור 15: תוצאת האיכון.

איור 15 מתאר את תוצאות תהליך איכון המקרה שתואר באיור 12. a הוא חזרה על איור 12 בתוספת הפתרון לבעיה המוצגת ו-b הוא ההגדלה של האזור הרלוונטי מ-a של איור 15, המדגיש את התוצאות.

הריבוע האדום מסמן את מיקומה האמיתי של המטרה, הפלוס הירוק מסמן את הניחוש הראשוני, הפלוס הורוד מסמן את המיקום המשוערך והאליפסה מגדירה את השטח שבו אמורה להימצא המטרה בהסתברות 90%. ניתן לראות שהאליפסה מכילה את מיקומה האמיתי של המטרה.

העבודה [K2007] מציגה את הגישה המוקדמת יותר לפתרון בעיית המיקום וזאת שיטת Stansfield algorithm שהוצגה לראשונה ב-[S1947] ולאחר מכן הוסברה ביתר פרוט במאמרו של [A1958]. בשיטה זו מניחים שהשגיאות הן יחסית קטנות ו- $\phi_{ri} \approx \frac{p_i}{D_i}$ כאשר p_i הוא המרחק

בין נקודת ניחוש ראשונית לקוון שנקלט בחישן i. בשיטה זו, נקודת הניחוש הראשונית תהיה התוצאה הסופית. שיטה זו יותר פשוטה מבחינה חישובית, אבל הרבה יותר רגישה לאי-דיוקים. המודל שהוצג עד עתה מניח שהאיכון נעשה על המישור. בפועל, בחלק מהיישומים יש לקחת בחשבון את העקמומיות של כדור הארץ. למשל, קורנים שמשדרים בתחום רדיו ת״ג (HF) יכולים להיקלט ממרחק של מאות עד אלפי קילומטרים. ברור, שבמקרה זה לא ניתן להזניח את העקמומיות של כדור הארץ: הדרך הקצרה שמחברת שתי נקודות על הספירה היא הקשת של המעגל הגדול שעובר דרכן ולא הקו הישר שמחבר אותם. [G1995] מציג פתרון שבו (15) לוקח בחשבון את הספרה.

לסיכום, בהינתן מערך חישנים מגלי כיוון, ניתן לאכן מטרה שמשדרת אות אלקטרומגנטי. תוצר האיכון יהיה שיערוך מיקום המטרה, שזה בעצם פונקצית צפיפות גאוסיאנית שניתן לייצג אותה בעזרת התוחלת שלה ומטריצת השונות המשותפת. באפליקציות שונות משתמשים בצורה שקולה שלה שהיא הצגת אליפסה, כאשר מרכז האליפסה הוא ערך התוחלת של פונקצית השערוך, וגודל האליפסה והאוריינטציה שלה מוכתבים ע״י מטריצת השונות המשותפת. האליפסה מייצגת שטח שבו הסתברות הימצאות המטרה גבוהה מערך מסוים שהוגדר מראש (מקובל להשתמש באליפסה שמכילה את הקורן בהסתברות של 90%).

5 עקיבה אחר המטרה

5.1 הגדרת הבעיה

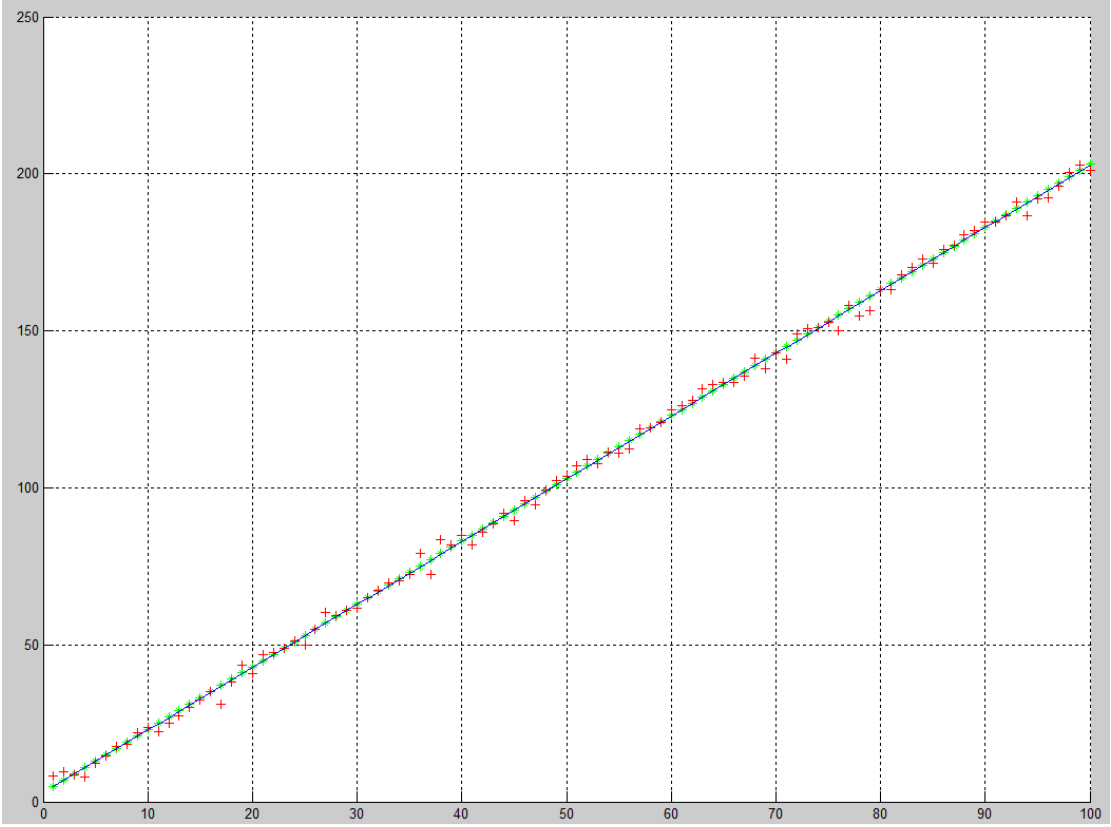
על מנת לעקוב אחר המטרה נדרש לקבל את אוסף המיקומים שלה לאורך הזמן. בפרקים הקודמים הוצגו אמצעי הקליטה – החישנים, והדרך לאכן את המטרה המתבססת על המדידות שנעשות במערך החישנים. האיכון שמתקבל אינו נתון מדויק, אלא פונקציה הסתברותית. פונקציה זו מגדירה את הצפיפות הגאוסיאנית הדו-ממדית של הסתברות הימצאות המטרה . דרך אחרת להצגה של האיכון היא אליפסת האי-ודאות כאשר מרכזה הוא המקום הסביר ביותר להימצאות של המטרה. סביר להניח, שלא תמיד מרכז האליפסה נמצא במקום הנכון.

איור 16: דוגמא למדידות של מטרה נעה.

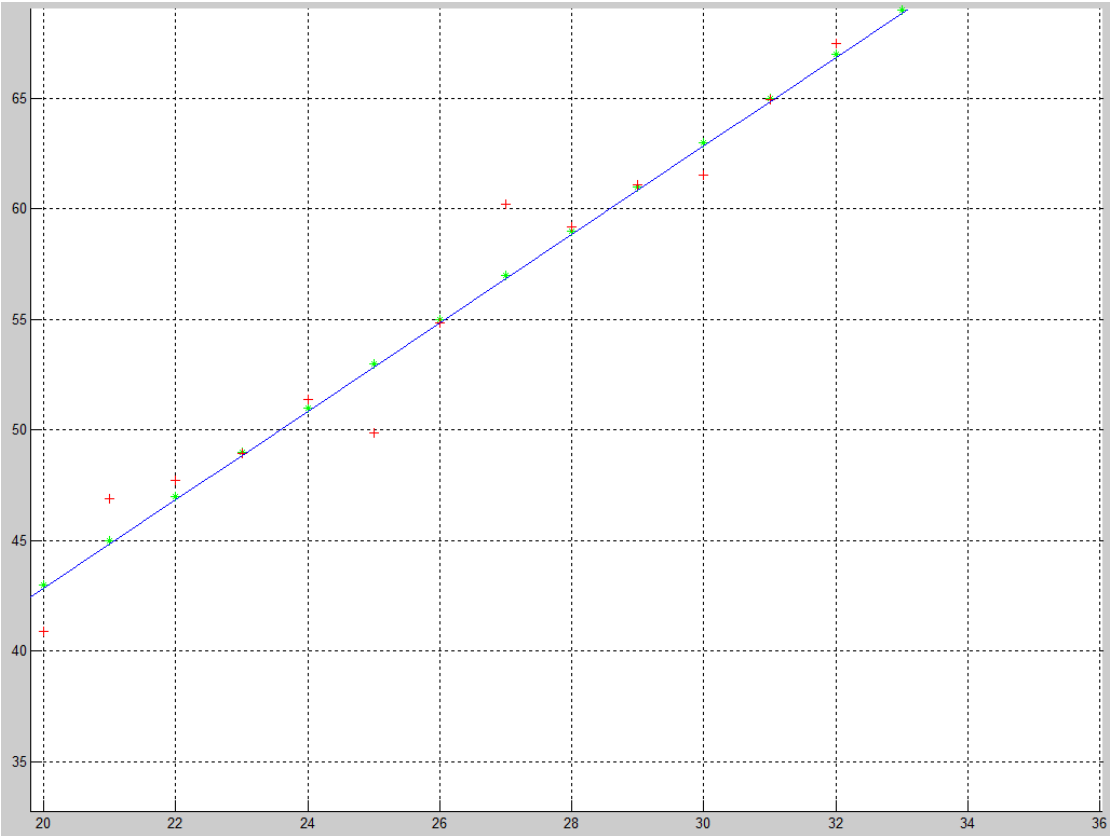
באיור 16 שנלקח מ http://pilotsfor911truth.org מתוארת דוגמא לעקיבה לאחר מטרות מוטסות. ניתן לראות באיור את המטרה שנעה לאורך הציר המסומן בכחול. מיקומה האמיתי של המטרה מסומן בנקודות ירוקות, האיכונים של המערכת מסומנים בנקודות לבנות. לכאורה, אפשר לחבר את הנקודות הלבנות בקו וזה יהיה המסלול של המטרה. קל לראות, שאומנם רוב האיכונים נמצאים בסביבת המטרה האמיתית, אך עדיין לא מדויקים. כפי שתואר בפרק המבוא, יש צורך לעקוב אחרי המטרה בצורה כמה שיותר מדויקת, על מנת להימנע מלקבל החלטות לא נכונות או מבזבוז המשאבים. במבוא הובאו מספר דוגמאות לשימוש בעוקב. אם הטיל המיירט יקבל כל פעם דיווחים לא מדויקים הוא יצטרך כל פעם לתמרן על מנת להתכוונן לעבר מיקום המטרה הצפוי. תמרון כזה יבזבז הרבה משאבים ובסוף הטיל המיירט יכול פשוט להחטיא. בדוגמת הבקרה האווירית, אם נתייחס רק למדידות שמתקבלות, עשויות להופיע התראות שווא על חריגה מהמסלול המוגדר או כניסה לאזורים אסורים. ריבוי התראות שווא יכול לגרום לירידה משמעותית בביצועים של מערך הבקרה. נדרש פתרון שיבטל ככל האפשר את הרעשים שנובעים

מאי דיוקים במדידה ויביא תוצאה קרובה ביותר לאמיתית. אפשר להניח שאין סטייה קבועה (bias) וכל הנתונים מוסטים בצורה מקרית.

הפתרון שמקובל להשתמש בו במקרים כאלו הוא שימוש במשערך LSE (Least Square Error) [K1993], [BD2001] שמביא פתרון המקטין את השגיאה הממוצעת של המדידות. אחד השימושים המוכרים הוא רגרסיה לינארית (linear regression) [K1993].



(a)



(b)

איור 17: דוגמא לרגרסיה לינארית.

הכוכביות הירוקות מסמנות את הנקודות האמיתיות של הפונקציה: $y = 2x + 3$. סימני הפלוס האדומים מסמנים את התוצאות ה"מורעשות" רעש לבן עם סטיית התקן של 2 $y_n = y + N(0,2)$. על מנת לשערך את הקו $y = kx + b$, משתמשים בנוסחת הרגרסיה הלינארית

$$k = \frac{\sum x_i y_i - \frac{1}{n}\sum x_i \sum y_i}{\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2}$$

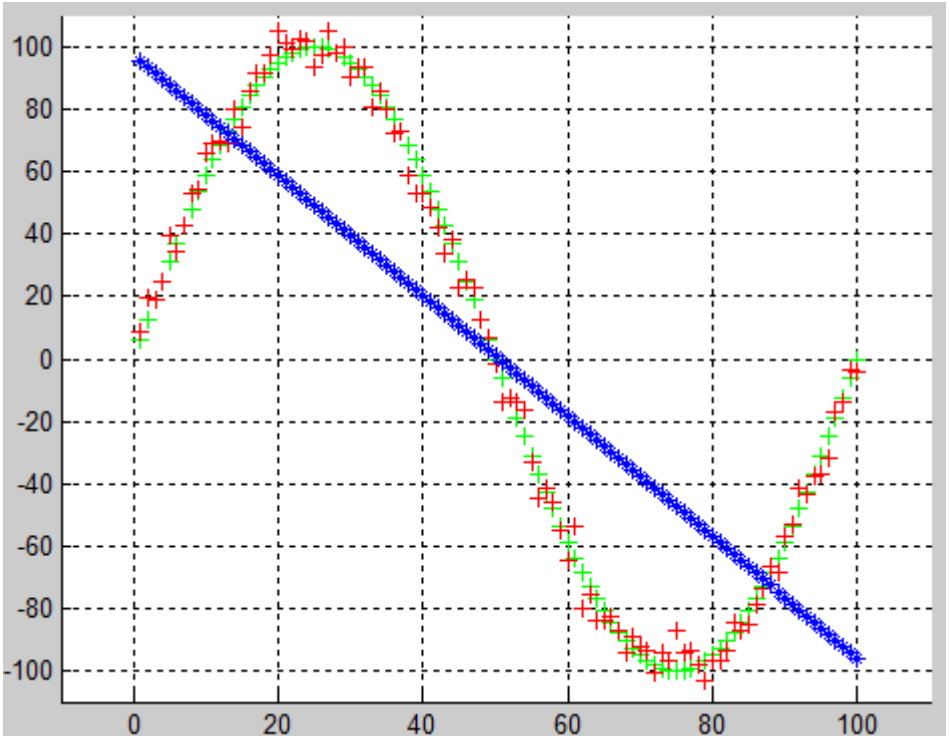
(43)

$$b = \bar{y} - k\bar{x}$$

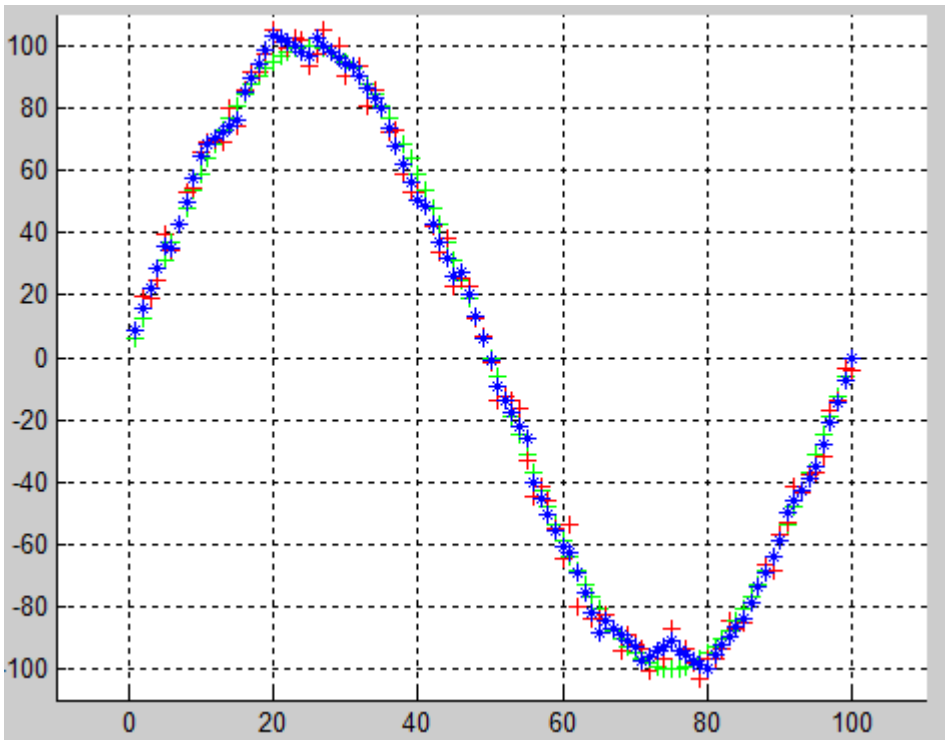
בדוגמא הנוכחית, $k = 1.9993, b = 2.8613$, הקו הכחול עובר קרוב מאוד לכוכביות הירוקות. ברור שככל שכמות הנקודות תעלה, כך גם התוצאה תהייה מדויקת יותר.

אומנם, הפתרון הזה מאבד מיעילותו כאשר המסלול לא יכול להיות מיוצג ע"י פונקציה אנליטית. במקרה כזה, אין טעם לחשב מקדמים של הפולינום אם לא קיים פולינום כזה. הפתרון האפשרי, לכאורה, הוא לחלק את כל המדידות לקבוצות ולעשות עליהן את הרגרסיה הלינארית. הפתרון כדוגמת הנ"ל אינו תמיד עובד. לדוגמא, אם הקבוצה של המדידות גדולה מדי, ומסלול המטרה

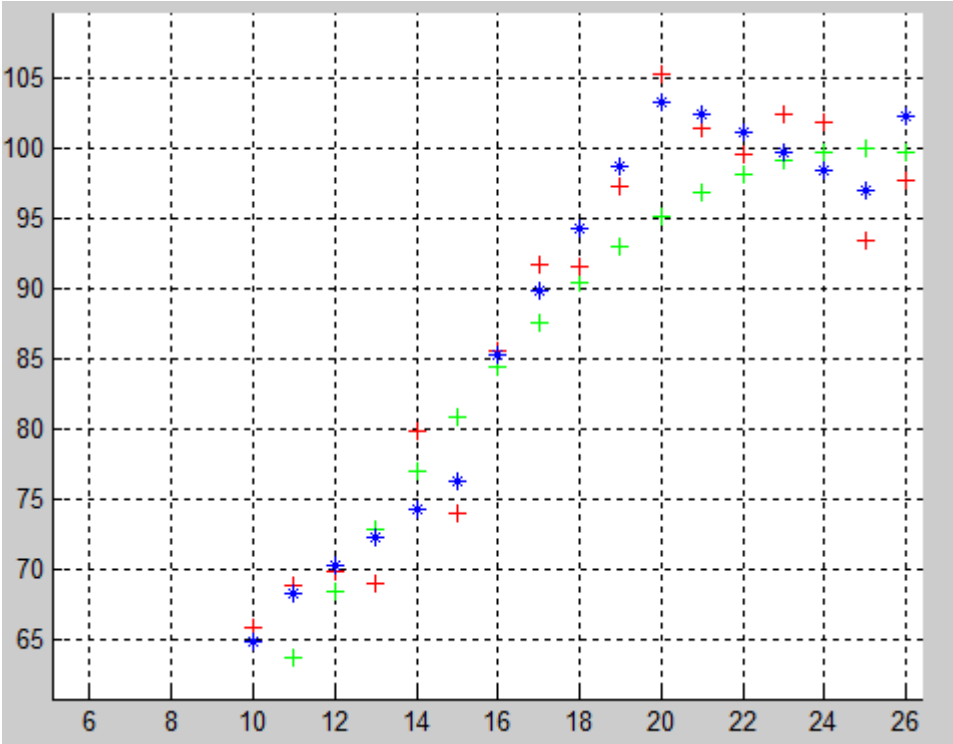
כולל תמרונים, אזי התוצאה שתתקבל תהיה מאוד לא מדויקת. מצד שני, אם הקבוצה שלעיל קטנה מדי, אזי התוצאה תהיה רגישה מדי לאי-דיוקים ושוב המסלול יהיה קופצני ולא מדויק.



איור 18: רגרסיה לינארית על כלל המדידות.



איור 19: רגרסיה לינארית על כל 5 מדידות.



איור 20: רגרסיה לינארית על כל 5 מדידות בהגדלה (zoom in).

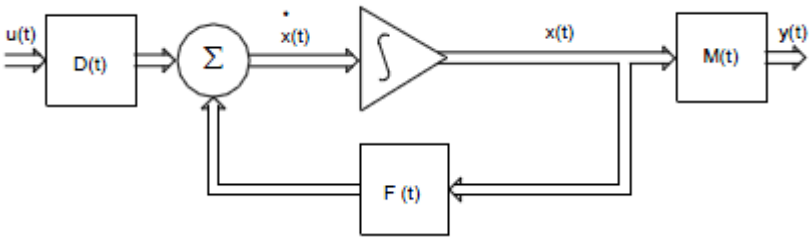
באיורים 18-20 סימני הפלוס הירוקים הם מיקומים אמיתיים, סימני הפלוס האדומים הם מדידות עם רעש, הכוכביות הכחולות מסמנות שערוך של המיקום ע"י רגרסיה. באיור 18 הרגרסיה נעשית על כל המדידות, כאשר באיורים 19 ו-20 הרגרסיה נעשית מחדש על כל חמש מדידות. איור 20 הוא הגדלה של חלק מאיור 19.

[K2007] מציג מספר שיטות להתגבר על הבעיה הזו כגון שיטות CPLE, Gauss-Newton, Recursive Least Squares ובסוף את שיטת Kalman Filter. שתי השיטות הראשונות עובדות בצורה של אצווה, כלומר חישוב המסלול נעשה בדיעבד לאחר קבלת כל המדידות. שיטות RLS ו-KF עובדות בצורה רקורסיבית, כלומר הן מתעדכנות תוך כדי קבלת הנתונים וכך מתאפשר לנצל אותם ביישומי זמן אמת. לפי [K2007] שיטת RLS סובלת מבעיות יציבות בהרבה מקרים, לכן בהמשך נתמקד בפתרון של Kalman Filter.

5.2 *הפתרון של Kalman Filter*

[K1960] מטפל במערכות דינמיות בדידות, כלומר, משתנות עם הזמן ונדגמות במרווחים שווים. הוא קובע שכדי לתאר מערכת דינמית, מספיק לתאר את המצבים של המערכת. כמו כן, מניחים שהמערכת היא לינארית ושהקלט של המערכת הוא תהליך מקרי גאוסיאני. הנחה זו מתבססת על העובדה שבהינתן טרנספורמציה לינארית $y = T(x)$, אם x הוא גאוסיאני, אזי גם y יהיה גאוסיאני.

מטרתו של המאמר לטפל במערכת מהסוג הבא:



איור 21: מערכת לינארית דינמית.

איור 21 נלקח מ- [K1960].

כך שאי-הדיוק יהיה מינימלי.

u(t) – קלט המערכת

x(t) – מצב המערכת

y(t) – פלט המערכת

D(t), F(t),M(t) – התמרות בתוך המערכת. אם שלושתן קבועות, אזי המערכת אינה תלוית זמן.

תאור המערכת המוצע הוא

$$\begin{cases} \dfrac{dx}{dt} = F(t)x + D(t)u(t) \\ y(t) = M(t)x(t) \end{cases} \qquad (44)$$

בסופו של דבר המאמר מגדיר את הבעיה בצורה הבאה:

נתונה מערכת דינמית מסוג

$$x(t+1) = F(t)x(t) \qquad (45)$$

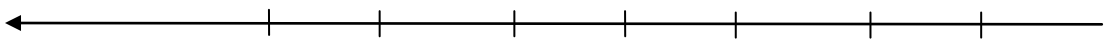
$$y(t) = M(t)x(t) \qquad (46)$$

ומתקבלות מדידות פלט של המערכת: $y_1, y_2, ..., y_n$. מה שנדרש הוא לשערך את מצבי המערכת $x_1, x_2, ..., x_n$ שהכי יתאימו לפלטים של המערכת.

נציג עכשיו דוגמא שתסביר בצורה טובה יותר את הנ"ל.

נאמר שאנחנו רוצים לתאר תנועה של רכב שאמור לזוז בקו ישר. אנחנו מודדים את המיקום שלו כל שנייה.

S



אומנ t_0 רק הנ t_1 זרצוי t_2 ל מר t_3 הרכ t_4 משר t_5 קרי t_6 לג גאוסיאנית

$N(0,4)$.

נגדיר את מצב המערכת כוקטור מורכב מהמיקום והמהירות של הרכב $x = \begin{pmatrix} s \\ v \end{pmatrix}$. אם ידוע ש-

$x_t = \begin{pmatrix} s_t \\ v_t \end{pmatrix}$, אז $x_{t+1} = x_t + \begin{pmatrix} v_t * 1 \\ n_v \end{pmatrix} = \begin{pmatrix} s_t + v_t * 1 \\ v_t + n_v \end{pmatrix}$; n_v נקרא רעש המודל. מצד שני כל שנייה עושים מדידות של מיקום הרכב שגם להם יש טעות מדידה שמתפלגת גאוסיאנית $N(0,1)$. טעות זאת נקראת גם רעש המדידה.

נכתוב מחדש את הנוסחאות (45),(46) תוך התחשבות ברעשים

$$x_k = Fx_{k-1} + w_{k-1} \tag{47}$$

$$y_k = Mx_k + v_k \tag{48}$$

$$p(w) \sim N(0, Q) \tag{49}$$

$$p(v) \sim N(0, R) \tag{50}$$

כאמור, בעקבות המדידות מתקבלים הנתונים $s_1, s_2, ..., s_n$. מה שנדרש לגלות הוא מה היו מצבי המערכת האופטימליים שהיו מתאימים למדידות אלו, כלומר השגיאה המצטברת בין המצבים אל המדידות המתאימות תהיה מינימלית.

הפתרון שהוצע ע"י [K1960] והוסבר גם במאמרים של [WB2006] ו-[M197] הוא:

נגדיר $\hat{x}_k^-$ כמצב apriori בשלב k, כלומר שערוך מצב המערכת לפי הידע שהיה לפני קבלת המדידה y_k ו- $\hat{x}_k$ כמצב aposteriori בשלב k, כלומר המצב המשוערך אחרי טיפול במדידה y_k. כמו כן, נגדיר שגיאות שערוך apriori ו- aposteriori

$$\begin{aligned} e_k^- &\equiv x_k - \hat{x}_k^- \\ e_k &\equiv x_k - \hat{x}_k \end{aligned} \tag{50}$$

נגדיר גם את השונות המשותפת של שגיאת השיערוך apriori ו- aposteriori

$$\begin{aligned} P_k^- &= [e_k^- e_k^{-T}] \\ P_k &= [e_k e_k^T] \end{aligned} \tag{51}$$

אז,

$$\hat{x}_k^- = F\hat{x}_{k-1} \tag{52}$$

$$P_k^- = FP_{k-1}F^T + Q_{k-1} \tag{53}$$

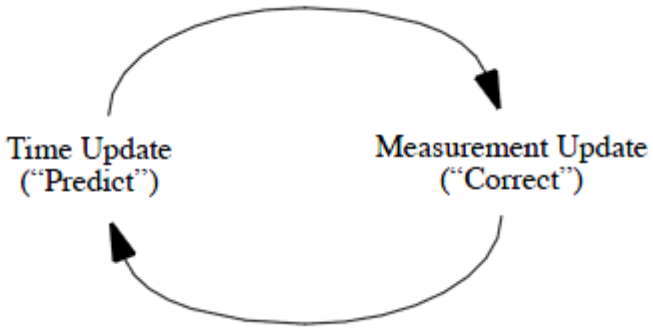
$$\begin{aligned} K_k &= P_k^- M^T (MP_k^- M^T + R_k)^{-1} \\ &= P_k^- M^T S_k^{-1} \end{aligned} \tag{54}$$

$$\hat{x}_k = \hat{x}_k^- + K(y_k - M\hat{x}_k^-) \quad (55)$$

$$P_k = (I - K_k M)P_k^- \quad (56)$$

ההפרש $r_k = (y_k - M\hat{x}_k^-)$ נקרא innovation והוא מייצג את ההפרש בין המדידה למצב המשוערך לפי המודל: ככל שערך זה קטן יותר, כך התנהגותה של המערכת תואמת את הציפיות. הערך K נקרא Kalman gain. משמעותו היא להגדיר עד כמה ה- innovation אמור להשפיע. אפשר לראות את ההיריסטיקה: ככל ש- R גדול יותר, כלומר האי-ודאות של המדידה גדולה יותר, אזי המדידה הזו תשפיע פחות על השערוך של $\hat{x}_k$. נבחן מקרה פרטי כאשר שגיאת המדידה שואפת ל- 0, אז $\lim_{R\to 0} K = M^{-1}$ ואז $\hat{x}_k = y_k$, כלומר, מצב המערכת חייב להיות שווה למדידה. האינטואיציה של המקרה היא ברורה. אם המדידה מושלמת (נטולת שגיאות), אזי היא צריכה להתאים למצב המערכת. לעומת זאת, אם מטריצת השונות של שגיאת שערוך ה-apriori, שואפת ל- 0, אז $\lim_{P_k^- \to 0} K = 0$ ואז $\hat{x}_k = \hat{x}_k^-$, כלומר, מצב המערכת יהיה כפי שהוא שוערך apriori. שוב האינטואיציה היא פשוטה: מכיוון שמערכת היא מדויקת, אזי הפרדיקציה של המצב מנחשת אותו במדויק.

אפשר להגיד שהמנגנון עובד בשני שלבים:



איור 22: מחזור העבודה של Kalman Filter [K1960].

איור 22 נלקח מ- [K1960].

1. ניחוש – prediction (52)-(53)
2. תיקון - correction (54)-(56)

[SBC1999] מביא פיתוח של ה- Kalman Filter בצורה כללית, כאשר הוא מתייחס גם למדידות וגם למצב המערכת כפונקציות רציפות המתארות את צפיפות ההסתברות. הגדרת הבעיה שלו אומרת מה היא ההסתברות להיות במצב מסוים ולקבל מדידה מסוימת בהינתן שהמדידה אכן התקבלה. גם פה שיערוך המצב aposteriori נעשה בעזרת שימוש ב- MLE על פונקציית התפלגות גאוסיאנית.

אם נחזור לבעיה של הרכב, שתוארה בתחילת פרק זה, נצטרך להגדיר את המודל על מנת להשתמש באלגוריתם של Kalman Filter. F תהיה מטריצה 2x2

$$F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix}$$

נזכור, שאת המטריצות F ו- M מכפילים בוקטור המצב X. כעת, ניתן לקבל מדידות y, ולהתחיל לשערך את מצבי המערכת x בצורה האופטימלית.

במקרה של עקיבה אחרי מטרות מוטסות יש להגדיר בהתאם את וקטור המצב: למשל, הוא יכול לכלול 6 ערכים – 3 קואורדינטות ו-3 מרכיבי מהירות. כמו כן, יש להתאים את מודל התנועה ושגיאות המודל בהתאם לציפיות. איכות העקיבה תלויה בצורה מוחלטת באיכות המודל שנבחר.

5.3 שיפורים והרחבות של Kalman Filter

5.3.1 הפתרון מסוג Adaptive Kalman Filter

לשיטה זו יש מספר מגבלות. המגבלה הראשונה היא שנדרש להגדיר את המודל בצורה קשיחה והיא אמורה לענות על כל הצרכים שיתעוררו במהלך עבודת המערכת. יתכן אומנם שיש צורך לעקוב בעזרת אותו מודל אחרי מטוס קל שזז במהירות של כ- 300 קמ״ש, מטוס נוסעים עם מנוע סילון של חברות תעופה שזז מהירות של כ- 800 קמ״ש בנתיבים בינלאומיים, ומטוס קרב שמהירותו יכולה להגיע למעל 1000 קמ״ש וכמו כן הוא צפוי לתמרן. ברור, שהמודל שבנוי עבור מטוס קל עלול לתת ביצועים חלשים בעקיבה לאחר מטוס קרב.

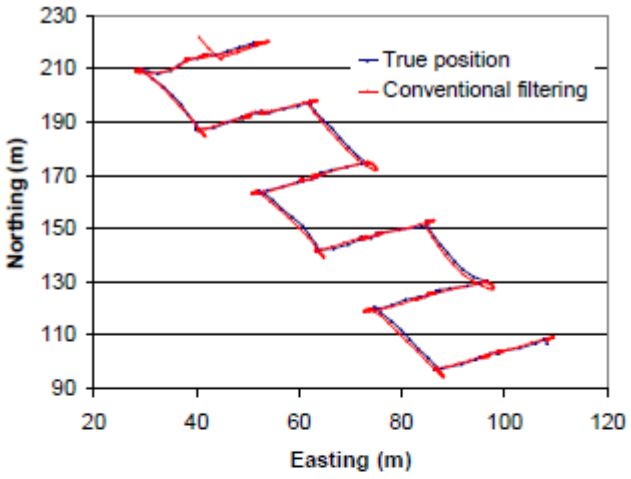
[HCCL2003] מציע להשתמש ב- Adaptive Kalman Filter, כלומר, המודל ישתנה כתלות במדידות. בפועל במאמר מוצעות שתי שיטות:

1. שיטת הזיכרון הדועך. משנים את (53) לצורה הבאה:

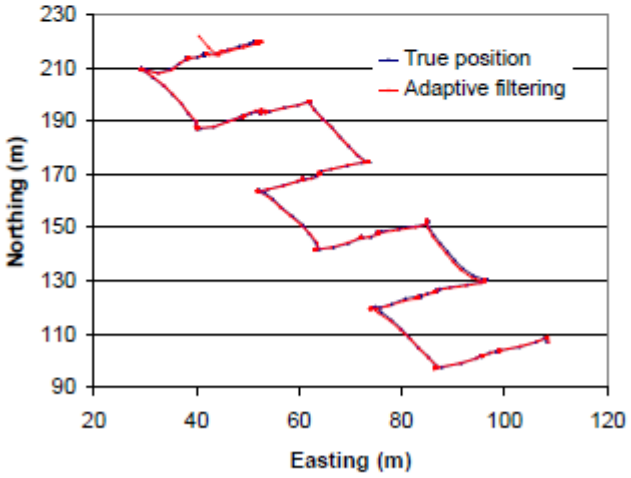
$$P_k^- = S(FP_{k-1}F^T + Q) \qquad (57)$$

אם S=1 , זאת השיטה הרגילה, לעומת זאת, אם $S > 1$, אז יש יותר אי-ודאות בשגיאת השיערוך ולכן למדידות יש יותר ״משקל״ להשפיע. ברור שהמצב שבו לפרמטר S יהיה ערך קבוע, יהיה יחסית לא יעיל ולכן הוצע אלגוריתם לחשב את הפרמטר S המתבסס על המרחק בין המצב ה-aposteriori למדידה בפועל, כך שבכל שלב הערך של S ישתנה בהתאם לטיב המודל.

2. שיערוך מרכיב השונות של המערכת הדינמית.



(a)Conventional Kalman filter



(b)Adaptive Kalman filter

איור 23: השוואת ביצועים בין האלגוריתמים.

איור 23 נלקח מ- [HCCL2003].

ניתן לראות שביצועים של האלגוריתם האדפטיבי טובים יותר, במיוחד כאשר הרכב מבצע פניות חדות. מודל רגיל, שמצפה שהתנועה תהיה לינארית, לא מצליח מספיק מהר להסתגל לשינוי חד. בשיטה השנייה הסתגלות זו מהירה יותר.

קיימת שיטה נוספת לעדכון של המודל [EM2001]. בשיטה זו משתמשים בטכנולוגיה של Fuzzy Logic. בטכנולוגיה זו מתמודדים עם פונקציה רציפה שמתארת את מידת קיום תכונה מסוימת. לדוגמא, ניתן להגדיר תכונות כגון גדול/קטן, חזק/חלש מבלי לקבוע סיפים שקובעים עבור כל ערך לאיזו קטגוריה שייך נושא ספציפי. ניתן לקבוע חוקים מסוימים שעל פי הם מסיקים מסקנות שגם הן בסך הכול פונקציות רציפות שמתארות תכונה כלשהי. לבסוף נעשה תרגום למספר (defuzzification).

[EM2001] מנסה לעדכן את שגיאת המדידה בעזרת הטכנולוגיה של Fuzzy Logic. הרעיון הוא לבדוק את מידת ההתאמה של שערוך שגיאה של innovation התאורטי ששווה ל- S_k לשערוך השגיאה הנמדדת לאורך העבודה של העוקב. לצורך החישוב משתמשים בחלון של מספר מצבים ומחשבים בצורה הבאה:

$$C_{rK}^{\wedge} = \frac{1}{N}\sum r_i r_i^T \qquad (58)$$

לפי המאמר בודקים את מידת ההתאמה ובהתאם משנים את R בצורה הבאה:

- אם $S_k = C_{rK}^{\wedge}$, אזי לא משנים את R
- אם $S_k > C_{rK}^{\wedge}$, אזי מקטינים את R, כלומר למדידה יש יותר משקל
- אם $S_k < C_{rK}^{\wedge}$, אזי מגדילים את R, כלומר למדידה יש פחות משקל

במילים אחרות, בודקים בפועל עד כמה הפרדיקציה טובה ומשערכים את השגיאה. אם השגיאה קטנה בפועל מהתכנון התאורטי, אז מקטינים גם את שגיאת המדידה, כדי שהמודל והמדידה יהיו באותם קני מידה. כנ"ל גם במקרה שהשגיאה גדולה מהתכנון, אז גם מגדילים את האי-הודאות של המדידה.

5.3.2 הפתרון מסוג Extended Kalman Filter

מגבלה נוספת של שיטת Kalman Filter היא ההנחה שהמודל הוא לינארי. בחיים אמיתיים יש הרבה מקרים שההנחה הזאת לא מספיק מדויקת. לכן, הוצעה השיטה שנקראת Extended Kalman Filter (EKF).

הרעיון של ה- Kalman Filter בנוי על זה שבכל שלב יודעים את פונקצית הצפיפות של המצב ושל המדידה. זה מושג ע"י תכונת ה- sufficiency [BD2001] של התפלגות גאוסיאנית: התוחלת והשונות מגדירים את פונקצית הצפיפות בכל נקודה. מכיוון שהמערכת היא לינארית, אז ההתמרות על פונקציות גאוסיאניות נותנות במוצא פונקציה גאוסיאנית. לעומת זאת, אם המערכת תהיה לא לינארית, למשל אם F היא פונקציה לא לינארית, אז מה שיתקבל במוצא כבר לא יהיה גאוסיאני. לכן, על מנת לחשב את שני המומנטים הראשוניים יש לחשב בפועל את התוחלת והשונות של ההתפלגות וזו משימה קשה מבחינה חישובית [R2004]. השיטה המוצעת היא לעשות שיערוך של המצב האופטימלי בעזרת הלינאריזציה בסביבה של המצב האחרון ששוערך.

בשיטת הלינאריזיציה של פונקציה לא לינארית השתמשנו בעבר בשערוך האיכון

$$f(x) \cong f(x_0) + G(x - x_0) \qquad (9)$$

G היא מטריצת נגזרות חלקיות של פונקציות $f(x)$ לפי כל אחד מהמשתנים

$$G = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}\Big|_{x=x_0} & \cdots & \cdots & \frac{\partial f_1}{\partial x_N}\Big|_{x=x_0} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial f_N}{\partial x_1}\Big|_{x=x_0} & \cdots & \cdots & \frac{\partial f_N}{\partial x_N}\Big|_{x=x_0} \end{bmatrix} \tag{10}$$

התהליך שמוצע בשיטת EKF הוא הבא [R2004] – המשתנים כבר הוגדרו בנוסחאות (47)-(56):

1. נניח קיים מצב משוערך $\hat{x}_{k-1}$
2. נעשה לינאריזציה של $x_k = f(x_{k-1}) + w_{k-1}$ בסביבה של $\hat{x}_{k-1}$.
3. נחשב את $\hat{x}_k^-$ ואת P_{k-1}^- לפי פונקציה לינארית שהשגנו בשלב 2
4. נעשה לינאריזציה של $y_k = m(x_k) + v_k$ בסביבה של $\hat{x}_k^-$.
5. נבצע שלב של עדכון המסנן ובעזרת הפונקציות הלינאריות שהתקבלו נקבל את $\hat{x}_k$ ואת P_{k-1}

נגדיר

$$\widetilde{F}_k = \nabla f_k \mid \hat{x}_k \tag{59}$$

$$\widetilde{M}_k = \nabla m_k \mid \hat{x}_k^- \tag{60}$$

כלומר $\widetilde{F}_k$ הוא Jacobian של f_k ו- $\widetilde{M}_k$ הוא Jacobian של m_k.

נשכתב את הנוסחאות (52)-(56) תוך התחשבות בלינאריזציה

$$\hat{x}_k^- = F\hat{x}_{k-1} \tag{61}$$

$$P_k^- = \widetilde{F}_{k-1} P_{k-1} \widetilde{F}_{k-1}^{\ T} + Q_{k-1} \tag{62}$$

$$K_k = P_k^- \widetilde{M}_k^{\ T} (\widetilde{M}_k P_k^- \widetilde{M}_k^{\ T} + R_k)^{-1} \tag{63}$$

$$\hat{x}_k = \hat{x}_k^- + K(y_k - M\hat{x}_k^-) \tag{64}$$

$$P_k = (I - K_k \widetilde{M}_k) P_k^- \tag{65}$$

[R2004] מביא את הפיתוח של נוסחאות (61)-(65).

חשוב לציין שתי עובדות הבאות:

1. EKF – הוא אינו המשערך האופטימלי בשונה מ- Kalman Filter הרגיל.

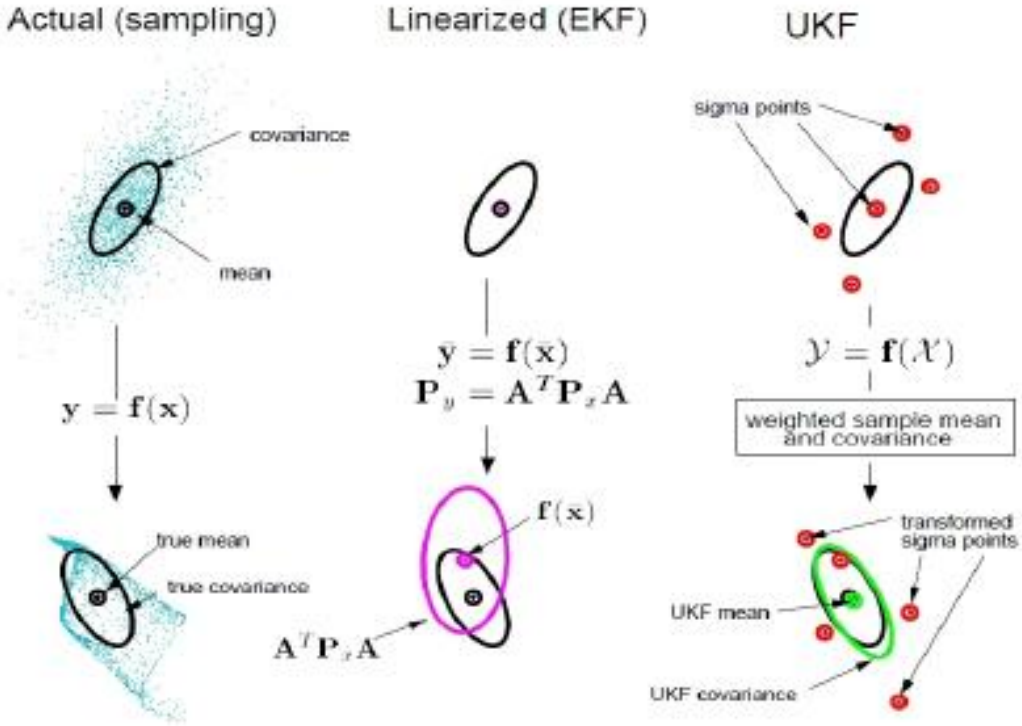
2. EKF – יכול להתבדר במקרים מסוימים, כלומר התוצאות שלו תהיינה שונות בצורה משמעותית מהערכים האופטימליים.

[BM2007] מביא מספר סיבות לעובדה 2 :

- לינאריזציה אמורה לעבוד רק אם ניתן לשערך את השגיאה בעזרת פונקציה לינארית.
- ההנחה היא שהמדידות מתפלגות פחות או יותר בצורה גאוסיאנית, אחרת חישוב התוחלת והשונויות לא יהיה נכון.
- לפונקציות התנועה חייב להיות קיים Jacobian, אם ישנן נקודות אי רציפות אז החישוב לא יהיה נכון.
- התכנסות המסנן תלויה בבחירה של המצב ההתחלתי ובפרמטרים של שגיאת המודל. במידה ולא יבחרו פרמטרים טובים החישובים לא יהיו נכונים.

5.3.3 הפתרון מסוג Unscented Kalman Filter

[BM2007] מתאר שיטה חלופית שנקראת UKF – Unscented Kalman Filter. לעומת השיטות שהוצגו עד כה שבהן פונקצית הצפיפות הוצגה בעזרת 2 מומנטים ראשוניים (התוחלת ומטריצת השונות), בשיטה זו בוחרים מספר נקודות מייצגות הנקראות sigma points. ההנחה היא שאם בוחרים את הנקודות נכון, ניתן לשערך בעזרתן מאפיינים של פונקציית הצפיפות. הרעיון של שיטת UKF הוא לעשות התמרה גם על הנקודות המייצגות ולשערך את המצב החדש בעזרת הנקודות שמתקבלות בעקבות ההתמרה.



איור 24: דוגמא להשוואה בין EKF ל- UKF.

איור 24 נלקח מ- [J2009]. באיור זה יש שלוש עמודות. בעמודה השמאלית למעלה רואים את המצב ה-(k-1). ניתן לראות את ההתפלגות בעזרת הנקודות הנדגמות וגם את אליפסת האי-ודאות. בעקבות הדינמיקה של המערכת עוברים למצב k. בחלק התחתון בעמודה השמאלית ניתן לראות לפי נקודות הדגימה שההתמרה הייתה לא לינארית. האליפסה מייצגת את התוחלת ואת השונות של הצפיפות המתקבלת. בעמודה המרכזית ניתן לראות איך EKF מתמודד עם הדינמיקה הנתונה. כתוצאה מכך מתקבלת אליפסה שאומנם חופפת עם האליפסה האמיתית אך בצורה חלקית בלבד וגם התוחלת המתקבלת די רחוקה מהתוצאה האופטימלית. המשמעות היא שבצעד הבא המרחק בין מה שהמצב של EKF מתאר למציאות יהיה עוד יותר גדול. בעמודה הימנית ניתן לראות את שיטת העבודה של UKF. בחלק העליון ניתן לראות 5 נקודות sigma points שמתארות את הצפיפות של מצב ה-(k-1). בחלק התחתון ניתן לראות את 5 הנקודות האלו שעברו את התמרה ואת האליפסה שמתקבלת כתוצאה מחישוב על הנקודות שהתקבלו. ניתן לראות שאליפסה שהתקבלה במקרה זה היא הרבה יותר דומה למה שהיה אמור להתקבל.

האינטואיציה של השיטה אומרת [BM2007] שיותר קל לעשות שיערוך להתפלגות מאשר לפונקציה לא לינארית מקרית כלשהי. לפי שיטה זו, כדי לשערך משתנה מקרי בעל n ממדים x_k נדרש לחשב 2n+1 נקודות בעלות משקל W לפי הנוסחא הבאה:

$$\chi^0{}_k = \hat{x}_k$$
$$W_0 = \frac{k}{(n+k)} \tag{66}$$

$$\chi_k^i = \hat{x}_k + (\sqrt{(n+k)_i P_k})_i \qquad \text{i=1,..,n} \tag{67}$$
$$W_i = \frac{k}{2(n+k)}$$

$$\chi_k^i = \hat{x}_k - (\sqrt{(n+k)_i P_k})_i \qquad \text{i=n+1,..,2n} \tag{68}$$
$$W_i = \frac{k}{2(n+k)}$$

בעקבות ההתמרה כל אחת מהנקודות עוברת בצורה הבאה:

$$\hat{\chi}^{i-}{}_k = F\hat{\chi}^i{}_k \tag{69}$$

$$\hat{x}_k^- = \sum_{i=0}^{2n} W_k^i \hat{\chi}^{i-}{}_k \tag{70}$$

$$P_k^- = \sum_{i=0}^{2n} W_k^i [\hat{\chi}^{i-}{}_k - \hat{x}_k^-][\hat{\chi}^{i-}{}_k - \hat{x}_k^-]^T \tag{71}$$

שלב העדכון שהוא בעצם החישוב של מצב $\hat{x}_k$ נעשה בצורה האיטרטיבית בשיטת Newton Raphson. מניחים

$$\bar{x}_0 = \hat{x}_k^- \tag{72}$$

מחשבים את $\bar{x}_j$ במספר איטרציות

$$\bar{x}_j = \bar{x}_{j-1} - (P_k^{-1} + M^T R^{-1} M)^{-1} * \left[P_k^{-1}(\bar{x}_{j-1} - \hat{x}_k^-) - M^T R^{-1}(y_k - m\bar{x}_{j-1})\right] \quad (73)$$

עד ש- $\bar{x}_j$ יהיה מספיק קרוב ל- $\bar{x}_{j-1}$, כלומר $\left|\bar{x}_j - \bar{x}_{j-1}\right| < \varepsilon$.

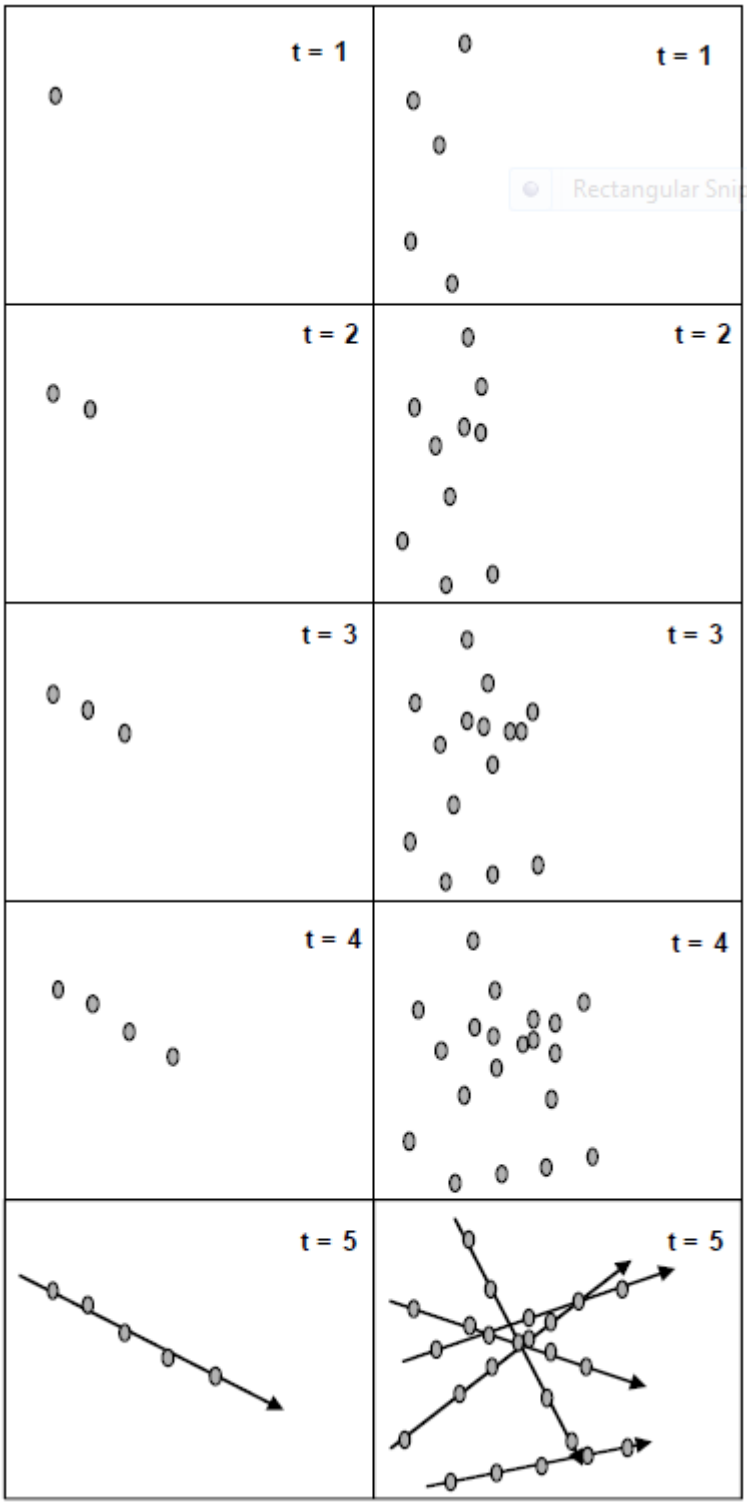
$$\hat{x}_k = \bar{x}_j \quad (74)$$

$$P_k = (P_k^{-1} + M^T R^{-1} M)^{-1} \quad (75)$$

[BM2007] מביא את פיתוח הנוסחאות הללו.

לסיכום, הצגנו עד עכשיו שיטות לעקיבה אחרי המטרה הבודדת אשר נעה במרחב. אומנם, הבעיה מתחילה להסתבך כאשר נדרש לעקוב אחר יותר ממטרה אחת. במציאות וגם בדוגמאות שהוצגו במבוא בדרך כלל יש מספר מטרות שנמצאות באזור העניין ונדרש לעקוב אחרי התנועה שלהן. בעיה זאת נקראת בספרות MTT (Multi Target Tracking).

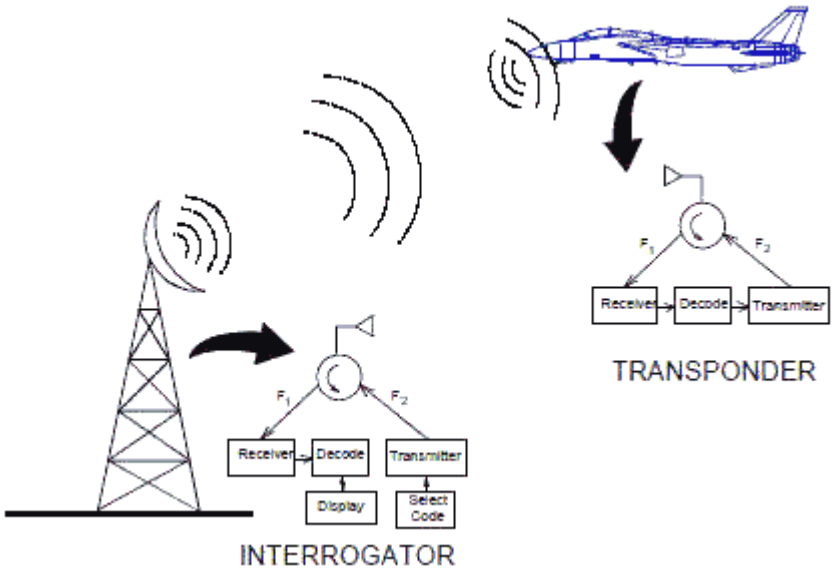
5.4 עקיבה אחר מספר מטרות - *Multi Target Tracking*



איור 25: דוגמא לעקיבה לאחר מספר מטרות.

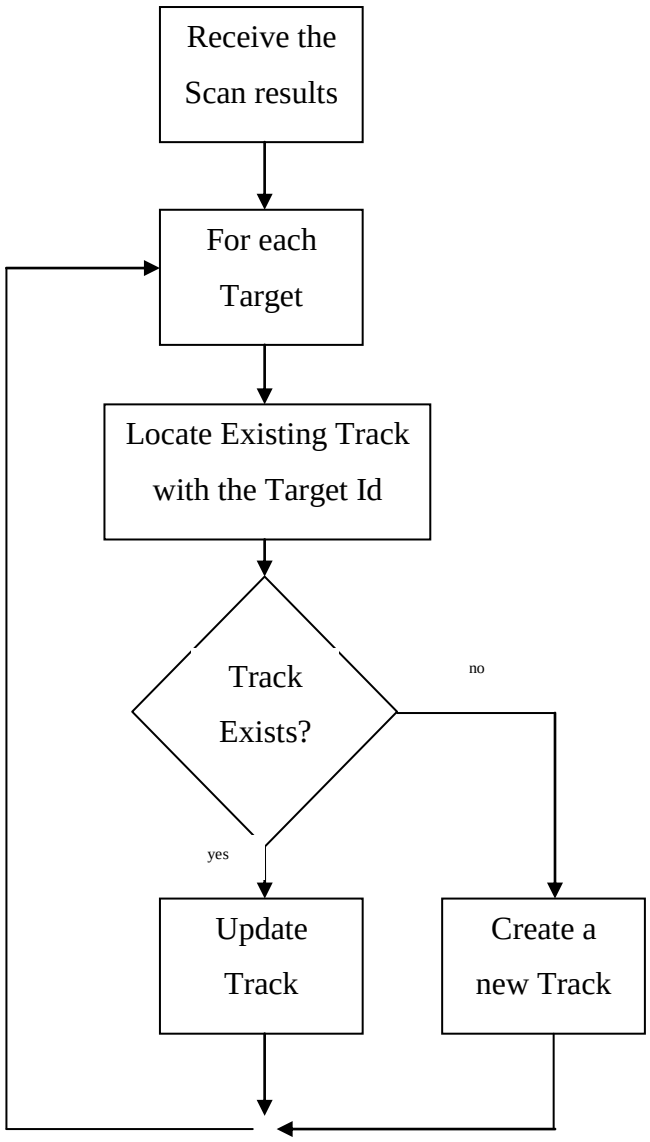
באיור זה, שנלקח מ- [HL2001] ניתן לראות בעמודה השמאלית את העקיבה אחר מטרה בודדת. בעמודה הימנית ניתן לראות אותה מטרה משתלבת עם עוד 4 מטרות שנעות במקביל. כל שורה מייצגת תמונה כפי שהיא מתקבלת בעקבות קבלת נתונים חדשים מהחישנים.

עולם הבעיה של MTT לפי [B2003] נראה כך: נניח שכבר קיימים במערכת מספר מסלולים (tracks) שנוצרו בעקבות מידע שהתקבל בעבר. כרגע מתקבלת קבוצה חדשה של מדידות מתוך מערך החישנים הקיים. באופן כללי, המדידות יכולות להתקבל במרווחי זמן קבועים שנקראים גם scans או data frames; או יכולות להתקבל בצורה מקרית. [B2003] משתמש במונח סריקה (scan) בהתייחס באופן כללי לכל קבוצה של מדידות החישנים שמתקבלות בו זמנית. המדידות של הסריקה החדשה יתחברו לתוך המסלולים הקיימים או יפתחו מסלולים חדשים.
המקרה הכי פשוט הוא מערכת IFF שנמצאת בשימוש בבקרות אוויריות.



איור 26: תאור סכמתי של מערכת IFF.

באיור 26 שנלקח מ http://rixco.multiply.com מתוארת דוגמה למערכת IFF. מערכת זו מורכבת מהמכ"ם המרכזי ה"מתשאל" (interrogator) שנמצא בבקרה האווירית ומשיבים (transponders) שנמצאים ברוב כלי טייס הקיימים כיום. האנטנה של המתשאל מסתובבת ושולחת בכל כיוון מסר "מי אתה". ברגע שהמשיב מקבל את הבקשה הוא מחזיר את התשובה שבה, בין השאר, הוא מביא את זיהויו של המטוס ביחד עם נתונים נוספים. המתשאל מקבל את המסר מהמטוס ומאכן את מיקומו בשיטות המכ"ם הרגילות כפי שתואר במבוא. כך, מערכת קרקעית מקבלת בכל סריקה את רשימת האיכונים החדשה של המטרות ביחד עם מספרי הזיהוי של כל מטרה. ניתן להפוך את הבעיה למקרה של עקיבה לאחר מטרה בודדת שמופעלת על כל מטרה בנפרד בצורה בלתי תלויה. מכיוון שלכל מטרה ומסלול יש זיהוי חד-ערכי, אפשר לתאר את האלגוריתם לעקיבה בצורה הבאה:

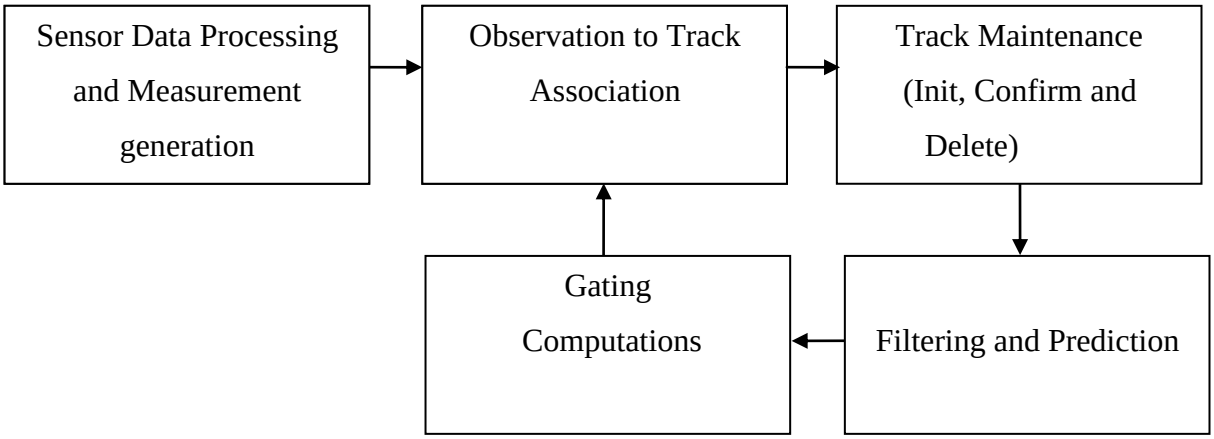


איור 27: תאור האלגוריתם לעקיבה לאחר מספר מטרות במערכת IFF.

1. מתקבלות תוצאות הסריקה שמכילות את מדידות המטרות שנקלטו (זיהוי+מיקום).
2. עוברים על כל מדידה בנפרד בצורה בלתי תלויה.
3. לכל מדידה מנסים למצוא את המסלול שלה על פי נתוני הזיהוי.
4. אם המסלול נמצא, מעדכנים על פי השיטות שתוארו בפרק שמטפל בעוקב לאחר מטרה בודדת.
5. אם המסלול לא נמצא, אז פותחים מסלול חדש ונותנים לו את הזיהוי שנמצא בתוך המדידה שכרגע מטפלים בה.

אומנם, רוב המערכות הקיימות לא מקבלות את זיהוי המטרה ולכן רדוקציה לבעיה של עוקב לאחר מטרה בודדת לא תמיד יעבוד.

התהליך הכללי ייראה בצורה הבאה:



איור 28: תאור מחזור תהליך MTT.

איור 28 מבוסס על [B2003]. בשלב הראשון מתקבלת סריקה מהחישנים. בשלב השני מנסים לשייך את המדידות למסלולים. בשלב זה מנסים למצוא זוג המורכב ממסלול ומדידה, כך שיתאימו בצורה מקסימלית לפרדיקציות שחושבו במחזור MTT הקודם.

בשלב השלישי מעדכנים את מבנה הנתונים של מסלולים הפעילים:

- מאשרים קיום של המסלולים הקיימים.
- פותחים מסלולים חדשים.
- מוחקים את המסלולים שהיו לא פעילים במשך זמן מסוים.

בשלב הרביעי מבצעים את עדכון המסלול בשיטות שתוארו קודם כגון Kalman Filter, כלומר מוצאים את $\hat{x}_k$ עבור מסלול זה, וגם מחשבים פרדיקציה $\hat{x}_{k+1}^-$ עבור הסריקה הבאה.

בשלב האחרון מחשבים את השער. משמעות השער הוא תת מרחב מסביב ל- $\hat{x}_{k+1}^-$ שבו יכולה להתקבל מדידה מהסריקה הבאה. אם הסריקה תתקבל מחוץ לתת מרחב זה, אז ההסתברות שהמדידה תתקבל מהמטרה שמתוארת ע"י מסלול זה נמוכה מאוד ואז שיוך כזה עלול להביא לשגיאה.

בתהליך העקיבה מתבססים על שתי הנחות:

1. הסתברות גילוי המטרה P_T היא גבוהה, כלומר אם המטרה נמצאת בטווח הקליטה של החישנים היא תתגלה ברוב הסריקות ותאוכן פחות או יותר במקום האמיתי. בכל זאת, הסתברות הגילוי אינה 1, ותהיינה סריקות שבהן המטרה עלולה לא להתגלות.
2. ישנה הסתברות להתראות שווא P_F, כלומר יתכן שבסריקה ידווחו איכונים של מטרות שלא באמת קיימות. התראות שווא אלו יכולות לנבוע מאיחוד של כמה מטרות קרובות ביחד, החזרים מעצמים נייחים כגון סלעים ובניינים, הפרעות וכו'.

מחזור חיי המסלול מורכב מ-3 שלבים/מצבים:

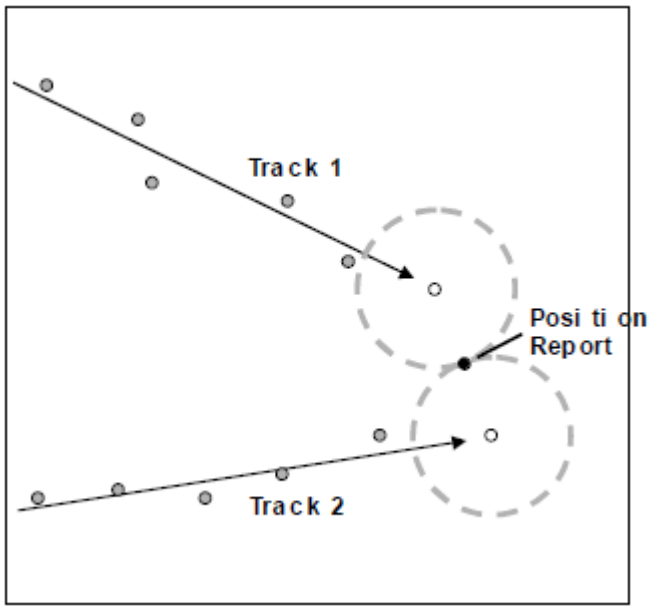
1. *מאותחל Initiated*. אם מתגלה איכון שלא משתייך לשום מסלול הוא פותח מסלול חדש במצב *מאותחל*. אומנם מסלול זה יכול להיות שגוי בגלל שהוא התבסס על התראת שווא או שגיאת השיוך.

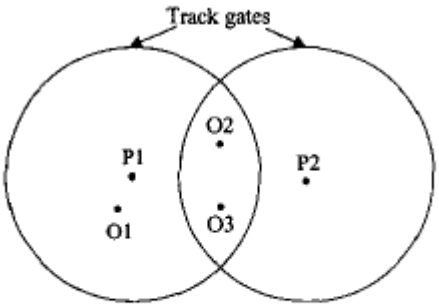
2. *מאושר Confirmed* . אם במשך כמה סריקות מסלול מאותחל התעדכן אז הוא עובר למצב *מאושר* . [B2003] מציע להשתמש ב- 3 עדכונים מתוך 4 לאחר פתיחת המסלול כדי להעביר אותו למצב המאושר. הרעיון אומר שאם הסתברות התראת שווא היא P_F , אז בהנחה שסריקות הן בלתי תלויות, ההסתברות לקבל 3 התראות שווא היא $(P_F)^3$, מה שמקטין מאוד את ההסתברות של הופעתן ברצף. לדוגמא, אם הסתברות של שגיאה היא $P_F = 0.1$, אז ההסתברות של יצירת מסלול שגוי היא 0.001.

3. *נמחק Deleted*. אם במשך כמה סריקות רציפות המטרה לא נצפתה אז היא כנראה נמצאת מחוץ לאזור גילוי החישנים. [B2003] מציע להשתמש ב-4 עד 7 סריקות רציפות ללא עדכון המסלול כדי להעביר אותו למצב *הנמחק*. הרעיון שוב מתבסס על אי-תלות בין הסריקות. אם הסתברות הגילוי היא P_T , אז הסתברות של אי-גילוי היא $1-P_T$ ואז הסתברות חוסר גילוי מטרה קיימת במשך 4 סריקות היא $(1-P_T)^4$. לדוגמא, אם $P_T = 0.6$ אז ניתן לחשב לפי הנוסחה את ההסתברות שמטרה קיימת תיסגר. ההסתברות שווה ל-0.0256 אם משתמשים ב- 4 סריקות, ול-0.0016 אם משתמשים ב- 7 סריקות. ככל שהסתברות הגילוי תגדל, כך תקטן גם ההסתברות למחיקה השגויה של המסלול.

כאשר המטרות נמצאות יחסית קרוב אחת לשנייה יכולות להתעורר סתירות (conflicts), כגון שתי מדידות יכולות להיות מתאימות לאותו מסלול, או מדידה אחת יכולה להיות מתאימה לכמה מסלולים.

לדוגמא:





O1, O2, O3 = Observation positions
P1, P2 = Predicted target position

איור 29: דוגמא לסתירה בין המדידות.

באיור זה שנלקח מ- [B2003] מתואר המצב הבא:
בשלב לפני הסריקה הנוכחית קיימים שני מסלולים. עבור כל אחד מהם חושבו פרדיקציות לקראת סריקה נוכחית P_1 ו- P_2. המעגל מסביב לכל אחת מהפרדיקציות הוא השער של המסלול. בעקבות הסריקה הנוכחית התקבלו שלוש מדידות: O_1, O_2 ו- . מהאיור ניתן לראות ש- O_1 יכולה להתאים אך ורק ל- P_1. לעומת זאת O_2 ו- יכולות להתאים גם ל- P_1 וגם P_2. ולכן השיוך אינו חד-משמעי. [B2003] מתאר שתי שיטות מקובלות בעולם ה- MTT.

1. Nearest Neighbor – NN: בשיטה זו מחפשים עבור כל פרדיקציה של מסלול את המדידה , כך שהמרחק בין מרכיבי הזוג הנבחר יהיה מינימלי, כלומר נבחרת ההשערה הסבירה ביותר. גישה זו אינטואיטיבית וקלה למימוש, אבל סובלת ממספר חסרונות:
   a. היא תלויה בסדר בחירת הזוגות. יתכן שאם הסדר ישתנה, ישתנו גם השיוכים.
   b. סיבוכיות החישוב היא MxN כאשר M היא כמות המסלולים, ו- N היא כמות המדידות בסריקה. חישוב המרחק הוא פעולה לא כל כך זולה, במיוחד אם זה מרחק סטטיסטי כגון מרחק Mahalanobis שמוגדר

   $$d^2 = (x-y)^T \left(\sum_x + \sum_y\right)^{-1} (x-y)$$

   , כאשר $\sum_x$ ו- $\sum_y$ הן מטריצות שונות של x ו- y בהתאמה.
2. Joint Probabilistic Data Association -JPDA. בשיטה זו, במקום למצוא את ההשערה הטובה ביותר, מעדכנים את המסלול בעזרת סכום משוקלל של כל המדידות הנמצאות בטווח של שער. לכן יתכן שמדידה מסוימת תתרום לעדכון של כמה מסלולים, למשל O_2 ו- ישתתפו בעדכון גם של P_1 וגם של P_2. גישה זו עמידה יותר בפני התראות שווא אבל סובלת מבעית "התקבצות" (coalescence), כלומר אם יש שתי מטרות שנעות בנתיבים מקביליים קרובים יתכן שיאוחדו ביחד.

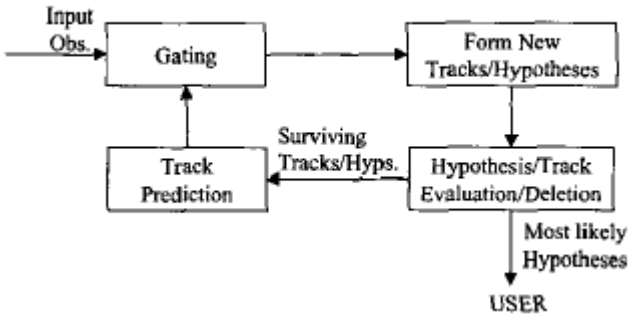
לסיכום, שיטת MTT נותנת פתרון לעקיבה אחרי מספר מטרות ע"י אלגוריתם איטרטיבי המבוסס על עוקב מסוג Kalman Filter. הביצועים של MTT מאוד תלויים מצד אחד, באיכות

החישנים, שמתבטאת בהסתברות גילוי גבוהה, והסתברות של התראות שווא קטנה. מצד שני, הביצועים תלויים גם בזירה עצמה. ככל שהזירה עמוסה, כך עולה הסיכוי לשיוך שגוי, אובדן מסלולים ובסופו של דבר הצגה של תמונת מצב שגויה.

5.5 הפתרון מסוג *Multi Hypothesis Tracking*

אחת הטכניקות שמתפתחת בשנים האחרונות לפתרון בעיית MTT היא שיטת MHT (Multi Hypothesis Tracking). הרעיון העיקרי של שיטה זו אומר את הדבר הבא: המגבלה העיקרית של השיטות שתוארו קודם היא שבמקרה של שיוך שגוי אי אפשר לחזור ולתקן. כפי שראינו, יש אפשרות לא מבוטלת של התראות שווא וגם טעיות באיכון, לכן טעות בהחלטה בודדת יכולה לגרום לשגיאה מצטברת בכל התמונה. בשיטת MHT ,שתתואר בהמשך הפרק, מחליטים סופית על השיוך של המדידה למסלול רק כאשר יש מספיק מידע כדי לקבל החלטה כזו[B2003].

[R1979] מתאר לראשונה את האלגוריתם שמבוסס על עבודות קודמות שטיפלו במעקב לאחר מטרה בודדת בסביבה של הפרעות clutter [SS1971].



איור 30: תאור של מחזור תהליך של MHT.

איור 30 נלקח מ- [B2003]. בשיטה זו מנהלים במקביל מספר השערות של שיוך המדידה למסלול, כלומר השערות אלו הן בלתי תלויות אחת בשנייה.

בשלב הראשון המדידות שמתקבלות מושוות למסלולים והשערות הקיימים במערכת. ההשוואות נעשות יחסית לשערים.

בשלב הבא נוצרים השערות ומסלולים חדשים על בסיס המדידות שהתקבלו.

לאחר מכן מנסים למחוק את ההשערות ואת המסלולים הלא תקפים.

בשלב האחרון משערכים פרדיקציות עבור כל ההשערות כהכנה למחזור הבא.

נשתמש באיור 29 כדי להמחיש את הנושא.

כאמור, בשלב לפני הסריקה הנוכחית קיימים שני מסלולים T_1 ו- T_2. עבור כל אחד מהם חושבו פרדיקציות לקראת סריקה נוכחית P_1 ו- P_2. בעקבות הסריקה הנוכחית התקבלו שלוש מדידות: O_1, O_2 ו- . מהאיור ניתן לראות ש-O_1 יכולה להתאים אך ורק ל- P_1. לעומת זאת, O_2 ו- יכולות להתאים גם ל-P_1 וגם ל- P_2. ישנן מספר אפשרויות: יתכן שהמדידות מתארות את המטרות הקיימות, או שכל שלוש המדידות מתארות מטרות חדשות, ויתכן שחלקן קיימות וחלקן חדשות.

נגדיר מסלול אפשרי $T_3 = (T_1, O_1)$, כלומר, T_3 מתקבל מעדכון של T_1 בעזרת O_1. בצורה דומה נגדיר $T_4 = (T_2, O_2)$ ו- $T_5 = (T_2, O_3)$. כמו כן, נגיד NT_1, NT_2 ו- NT_3 מסלולים חדשים כתוצאה מהמדידות O_1, O_2 ו- בהתאמה. ניתן לבנות 10 השערות אפשריות:

$$H_1 : T_1, T_2, NT_1, NT_2, NT_3$$
$$H_2 : T_3, T_4, NT_3$$
$$H_3 : T_3, T_5, NT_2$$

בהשערה H_1 כל המדידות החדשות יצרו מסלולים חדשים, לעומת זאת בהשערה H_2 המדידה O_1 התחברה ל- T_1, O_2 ל- T_2 ומדידה האחרונה יצרה מסלול חדש NT_3. המסלולים מוגדרים להיות *מתאימים* (*compatible*) אם אין להם מדידות משותפות. לדוגמא, המסלולים T_1 ו- T_2 הם מתאימים. ברור שלא יתכן ששתי מטרות תייצרנה בסך הכול מדידה אחת משותפת. בכל השערה עבור כל מסלול יש ציון שמתאר את טיבו. [B2003] מציע שיטה לחישוב הציון שיותר קלה חישובית מזו שהוצגה במקור ע"י [R1979]. השיטה של [B2003] מתבססת על כלל בייס ומגדירה את LR (likelihood ratio)

$$LR = \frac{p(D \mid H_1)P_0(H_1)}{p(D \mid H_0)P_0(H_0)} \stackrel{\Delta}{=} \frac{P_T}{P_F} \tag{76}$$

ההשערה H_1 היא ההשערה שהמדידה התחברה למסלול בצורה נכונה והסתברותה היא P_T וההשערה H_0 היא שהמדידה התחברה בטעות והסתברותה היא P_F.

$p(D \mid H_1)$ היא פונקצית צפיפות של ההסתברות לקבל מדידה D בהינתן שההשערה H_1 נכונה. בצורה דומה, $p(D \mid H_0)$ היא פונקצית צפיפות של ההסתברות לקבל מדידה D בהינתן שההשערה H_0 נכונה.

נגדיר גם

$$LLR = \ln[\frac{P_T}{P_F}] \tag{77}$$

ניתן בקלות לחשב הסתברות שהמדידה התחברה נכון בצורה הבאה:

$$\frac{P_T}{P_F} = \frac{P_T}{1 - P_T} = e^{LLR}$$
$$P_T = \frac{e^{LLR}}{1 + e^{LLR}} \tag{78}$$

הערך LLR הוא בעצם הציון של המסלול שמאפשר להשוות את טיב המסלולים הקיימים.

[B2003] מציעה שיטה רקורסיבית לחישוב ציון המסלול $L(k)$ בצורה הבאה:

$$L(k) = L(k-1) + \Delta L(k)$$
$$\Delta L(k) = \begin{cases} \ln(1-\hat{P}_D);\ no\,update \\ \Delta L_U(k);\ track\,updated \end{cases} \quad (79)$$

$\hat{P}_D$ ההסתברות הצפויה של גילוי המטרה בסריקה הנוכחית

$\Delta L_U(k)$ תוספת אמינות.

כדאי לשים לב, במקרה שמטרה לא מתגלית בסריקה, אז $\Delta L(k)$ יהיה שלילי ויקטין את הערך הכללי של המסלול $L(k)$.

על מנת להקטין את כמות ההשערות והמסלולים שגדלים בצורה מעריכית, משתמשים באיגוד (clustering), מחיקה של מסלולים והשערות ומיזוג של המסלולים.

הרעיון של האיגוד הוא להפריד בעיה לתת בעיות בלתי תלויות וכך להקטין את הסיבוכיות .

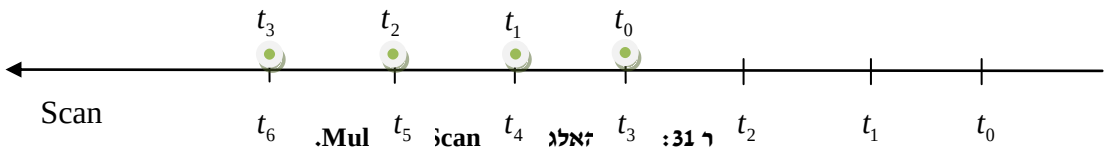
הרעיון אומר שאם $N = k*m$ ופונקציה f היא פונקציה כבדה חישובית $k*f(m) < f(k*m)$.

[B2003] מציע לקבץ לתוך אגד אחד את כל המסלולים שיש להם מדידות משותפות, אפילו בצורה עקיפה (טרנזיטיבית). אם אין להם מדידות משותפות הם יכולים להתנהל במקביל בצורה בלתי תלויה כמו בפתרון NN בשיטת ה-MTT.

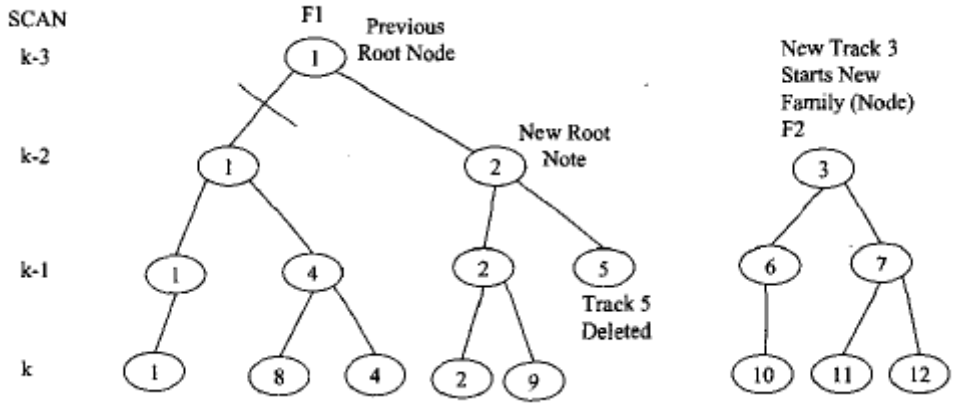
ישנם מספר אלגוריתמים למחיקת המסלולים והשערות. [R1979] מציג את השיטה שנקראת Multiple-Scan.

הרעיון העיקרי של השיטה הזו אומר, שאומנם כאשר מתקבלת הסריקה אין מספיק נתונים כדי לקבוע האם השיוך בין מדידה למסלול הוא טוב, אבל כעבור כמה סריקות אפשר להחליט בצורה הרבה יותר טובה. אינטואיטיבית, ההסתברות שההשערה לא נכונה תתקיים למשך N סריקות שווה ל- $(P_F)^N$, בגלל ההנחה שהסריקות הן בלתי תלויות. לעומת זאת ההסתברות שההשערה הנכונה לא תתקיים היא $(1-\hat{P}_D)^N$, כלומר שהמטרה האמיתית אף פעם לא התגלתה.

אפשר לתאר את האלגוריתם בציר הזמן בצורה הבאה:

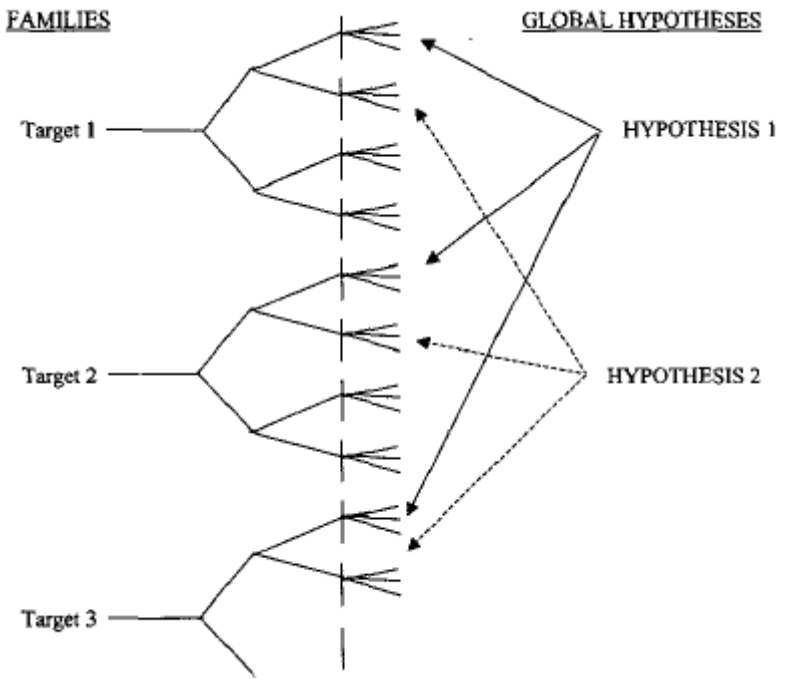


באיור 31 מתוארות סריקות שמתקבלות מ- t_0 עד t_6. בכל סריקה נבנות השערות עבור כל זוג אפשרי של מדידה ומסלול. כעבור N מחזורים, במקרה זה מחליטים לגבי השיוכים שנעשו בעבר. בנקודה t_3 אחרי כל השיוכים מחשבים את הציונים של כל המסלולים ובוחרים את המסלול הטוב ביותר ולפי זה מוחקים את השיוכים הלא נכונים שנעשו בשלב t_0. כנ"ל ב- t_4 מוחקים את t_1 וכן הלאה.



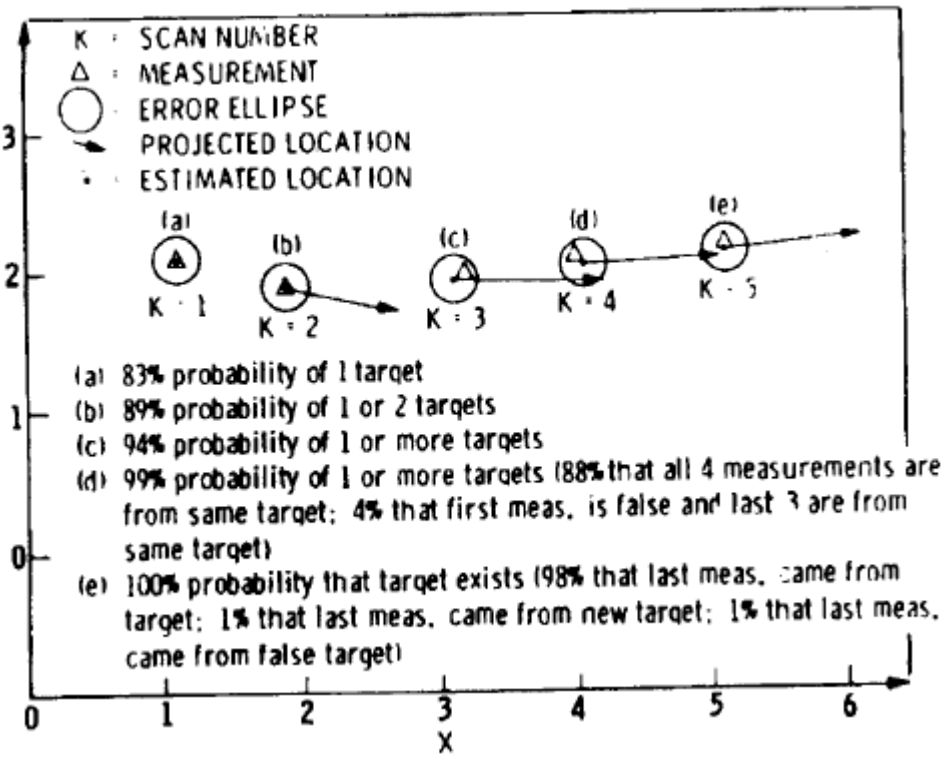
איור 32: תאור האלגוריתם Multiple Scan [B2003] .

ניתן לראות את התהליך הזה באיור 32 שנלקח מ- [B2003]. כאן משתמשים בגודל החלון N=2. נתמקד תחילה בחלק השמאלי של האיור ונראה את העץ F1. בתחילת סריקה k-3 קיים מסלול 1. במחזור הבא מתקבלת מדידה רלוונטית אחת לכן יש שתי השערות: או להמשיך לקיים רק את המסלול 1 כי המדידה מתחברת אליו או לפתוח מסלול חדש בנוסף 2. בצורה הבאה במחזור סריקה הבא יש כבר 4 אפשרויות: שני מסלולים קיימים וגם ניתן לפתוח עוד שני מסלולים נוספים. במחזור k מתקבלות 5 אופציות. בשלב זה צריך לקבל את החלטה לגבי השיוכים שנעשו בשלב k-N, כלומר שנעשו N מחזורים לפני המחזור הנוכחי k. נניח שלמסלול 2 יש ציון טוב ביותר. המשמעות של זה היא שתת עץ ששורשו הוא 1 ברמה של k-1 הוא שגוי, לכן מוחקים את כל התת-עץ שמכיל 5 צמתים. חשוב לציין, שלמרות ש-2 הוא הכי חזק, לא מוחקים את 9. ההחלטה לגבי 9 תבוא לאחר קבלת נתוני סריקה של k+2. בצד ימין של הציור אפשר לראות עץ נוסף F2 שמתקיים במקביל. עץ זה בעצם הוא משפחה (אגד) אחרת שלא הייתה יכולה להתחבר לצמתים של F1, כנראה בגלל שהם נמצאים מחוץ לשער. ההשערות של העץ נבנות בצורה בלתי תלויה מה שמקטין את כמות ההשערות ומקל על הסיבוכיות. מעניין לציין נקודה נוספת: אם N=0 אז מקבלים את האלגוריתם של NN בMTT, כלומר המדידה שמתקבלת מיד מתחברת למסלול הקרוב ביותר.



איור 33: תאור של השערות גלובליות [B2003] .

את הקשר בין ההשערות הגלובליות למשפחות. בדרך [B2003]ניתן לראות באיור 33 שנלקח מ- זאת יכול להיות נוח לדווח את מצב התמונה הנוכחית: יש לדווח למשתמש את ההשערה הגלובלית בעלת ציון הגבוה ביותר.



איור 34: דוגמא לאתחול מסלול חדש.

איור 34 נלקח מ- [R1979]. בדוגמא זו, שהובאה מתוך [R1979], מציגים את תוצאות הסימולציה כאשר $\hat{P}_D = 0.9$ ו- $P_F = 0.1$. באיור ניתן לראות את חישוב ההסתברויות והמסקנה שמתקבלת לאחר 5 סריקות שהמטרה אכן קיימת לכן המסלול מאותחל. [SBC1999] מתאר בצורה פורמלית את האלגוריתם MHT.

כפי שניתן לראות, האלגוריתמים MTT בכלל ו- MHT בפרט, מאוד כבדים מבחינה חישובית. הגורם שיכול להשפיע ולייעל משמעותית הוא מנגנון Gating, כלומר פסילה של מדידות לא רלוונטיות וביצוע שיוך בין מסלול רק לאחת המדידות שנמצאות בטווח שאליו מסוגלת להגיע המטרה. בדיקת Gating היא בעצם חישוב המרחק הרב-ממדי בין המסלול למדידה שכולל בתוכו גם חישובים חשבוניים לא פשוטים וגם עשוי לבצע היפוכי מטריצות. הבעיה מתחילה להיות קריטית במיוחד באזורים שיש בהם הרבה הפרעות (clutter). בשיטה הנאיבית יש לבצע MxN חישובי המרחק כאשר M היא כמות המסלולים, ו- N היא כמות המדידות בסריקה שכולל גם את התראות השווא שנובעות מההפרעות. במקרה של MHT, כמות ההשערות גדלה בצורה מעריכית, לכן יש חשיבות רבה להפריד ביעילות בין המשפחות השונות.

מהאמור לעיל עולה הצורך לבנות מבנה נתונים יעיל שיאפשר לצמצם במידת האפשר את כמות החישובים הנדרשים. הפרק הבא יתאר את נושא החיפוש ומבני הנתונים שאפשר להשתמש בהם.

6 מבני נתונים מרחביים

6.1 *ההגדרה הפורמלית*

בתהליך העקיבה עולות הרבה בעיות שדורשות פתרון יעיל. לדוגמא, נדרש למצוא את המסלול הקרוב ביותר למדידה. בעיה נוספת שעשויה להתעורר היא הצורך למצוא את כל המטרות הנמצאות במרחב מסוים. לעיתים הבעיה היא הפוכה, נדרש למצוא באיזה אזור נמצאת מטרה מסוימת על מנת להחליט מה הוא אופן הטיפול בה. ניתן להגיד בהכללה שיש צורך לטפל בבעיות החיפוש במרחב רב-מימדי. בפרק זה נתאר את בעיית החיפוש באופן פורמלי וניתן מספר דוגמאות לפתרון של חלק מן הבעיות.

באופן כללי, פעולת החיפוש היא אחת הפעולות שמופעלות על קבוצות דינמיות [CLR1990] . ניתן להגדיר שתי משפחות של פעולות שמופעלות על קבוצה דינמית:

1. פעולות עדכון (modifying operations) – שמשנות את הקבוצה, כגון פעולת *הכנסה* (Insert) ו*מחיקה* (Delete).
2. שאילתות (queries) – פעולות שמביאות מידע על הקבוצה מבלי לשנותה, למשל *חיפוש* (Search) , *מינימום*, *מקסימום* לקבוצות בעלות סדר מלא וכו'.

פעולת החיפוש Search (S,k) מוגדרת בצורה הבאה: שאילתה, בהינתן קבוצה S וערך מפתח k, מחזירה מצביע x לאיבר ב- S כך, ש- key[x] = k , או לחלופין, מחזירה nil אם לא קיים איבר כזה.

[AE1997] מגדיר את בעיית חיפוש בתחום בצורה הבאה:

נניח ש-S היא קבוצה דינמית בת n נקודות במרחב $\Re^d$ (בעל d ממדים). כמו כן נניח ש- R היא משפחה של תת-קבוצות של $\Re^d$. האיברים של R, כלומר התת-קבוצות של $\Re^d$ ייקראו תחומים (ranges). המטרה היא לבצע עיבוד מקדים על S כך שעבור תחום $\gamma \in R$ ניתן יהיה לספור ולדווח ביעילות את הנקודות של $S \cap \gamma$, כלומר את הנקודות של S שמתאימות ל- γ . בגאומטריה התחום יכול להיות קטע, מלבן, כדור, תת-מרחב וכו'. הפתרון הנאיבי של שאילתה ייקח זמן לינארי: יש לעבור על כל אברי S ולבדוק עבור כל אחד האם הוא שייך ל- γ . אומנם, אם נדרש לעשות שאילתות רבות, זמן הריצה הלינארי עבור כל שאילתה מתחיל להיות לא מספיק. יש לבנות מבנה נתונים שיאפשר לבצע שאילתה, הרצה בזמן מהיר יותר מלינארי.

[AE1997] מביא מספר דוגמאות לשאילתות תחום:

- לספור ו/או לדווח את כל הנקודות שנמצאות בתחום השאילתה.
- שאילתת קיום, כלומר לבדוק האם $S \cap \gamma = \varnothing$
- שאילתות אופטימיזציה, כלומר, אם רוצים לבחור נקודה בתחום שיש לה תכונה מסוימת, למשל למצוא נקודה בתחום γ בעלת קואורדינטת x_1 הגדולה ביותר.

נניח קיים מרחב S_. חבורה למחצה חלופית (+,S_) במרחב S_ הינה קבוצה $S \subseteq S\_$ שמוגדרת עליה פעולה $+: S\_ \times S\_ \to S\_$ כך שעבור כל האיברים $x, y \in S$ מתקיים ש- x+y=y+x. כמו כן, עבור כל נקודה $p \in S$ יוגדר $w(p) \in S\_$ שייקרא המשקל של הנקודה. נגדיר עתה את משקלה של קבוצה בצורה הבאה: $w(S') = \sum_{p \in S'} w(p),\ S' \subseteq S$, כלומר משקלה של התת-קבוצה S' הוא "סכום" המשקלים של הנקודות המרכיבות אותה. כעת נגדיר בצורה פורמלית את פתרון השאילתה על תחום $\gamma \in \Re$: $w(S \cap \gamma)$, כלומר, פתרון השאילתה יהיה משקלו של החיתוך בין קבוצת S המקורית לקבוצת γ. אם מטרת השאילתה היא לספור את כמות הנקודות בתחום γ, נגדיר חבורה למחצה $(\mathbb{Z},+)$ כאשר $\mathbb{Z}$ מייצג את קבוצת המספרים השלמים ופעולת + היא פעולת החיבור הרגילה. לכל p ניתן משקל w(p)=1. עבור כל נקודה שמתאימה ל-γ יתווסף 1 ל-$w(S \cap \gamma)$, לכן התוצאה הסופית תהיה כמות הנקודות שנמצאות בתחום γ.

עבור שאילתת קיום נגדיר חבורה למחצה $(\{0,1\},\vee)$, וניתן לכל אחת מהנקודות משקל 1. התוצאה של $w(S \cap \gamma)$ תהיה OR לוגי של כל ה-1-ים שיתקבלו מהנקודות שנמצאות בתחום γ. אם אין נקודות בכלל יתקבל 0, אחרת 1.

עבור שאילתת דיווח על הנקודות בתחום נגדיר חבורה למחצה $(2^S,\cup)$, ומשקלה של כל נקודה תהיה w(p)=p. התוצאה של $w(S \cap \gamma)$ תהיה איחוד כל הנקודות שנמצאות מתאימות לתחום γ.

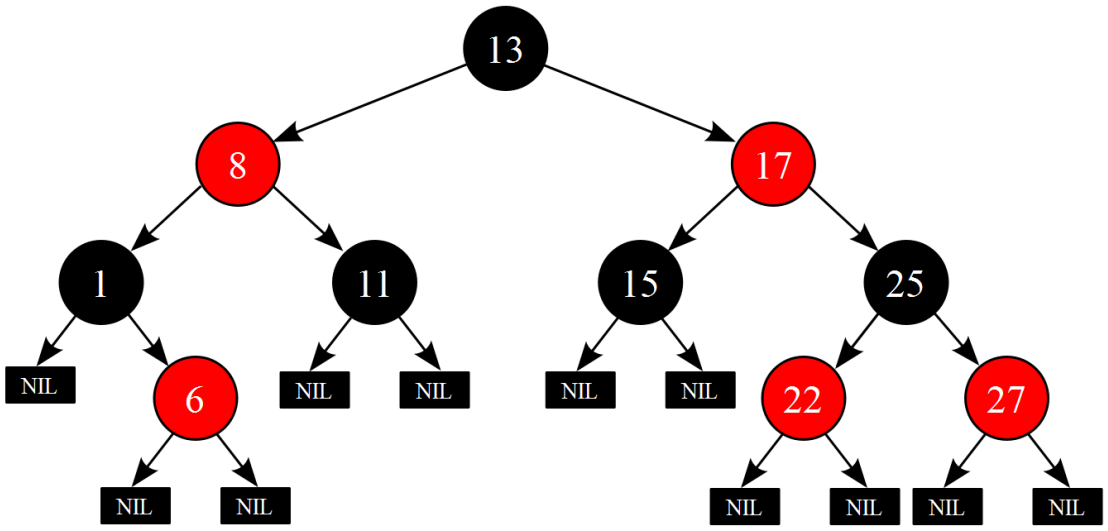
עבור שאילתת האופטימיזציה נגדיר חבורה למחצה $(\Re, \max)$, ומשקלה של כל נקודה תהיה w(p)= $x_1(p)$. התוצאה של $w(S \cap \gamma)$ תהיה הערך המקסימלי של קואורדינטת x_1 של הנקודות שיתאימו לתחום γ.

ניתן להגדיר באופן כללי את בעיית החיפוש הגאומטרי בצורה הבאה: נניח ש- S היא קבוצת העצמים ב- $\Re^d$ (נקודות, תת מישורים וכו'), $(+,\mathbf{S\_})$ היא חבורה למחצה חלופית, $w: S \to S\_$ היא פונקציית המשקל, R היא קבוצת התחומים ו- $\Diamond \subseteq S \times R$ הוא היחס המרחבי בין העצמים לתחומים. במקרה כזה מטרת השאילתה היא לחשב את $\sum_{p \Diamond \gamma} w(p)$. במקרה של חיפוש תחום הגאומטרי $\Diamond = \in$.

הביצועים של מבנה הנתונים נמדדים ב-3 פרמטרים:

1. זמן השאילתה – הזמן שעולה לבצע את השאילתה. זמן זה מורכב משני מרכיבים:
   a. זמן החיפוש – חיפוש של האיברים שמתאימים לתחום שתלוי בכמות האיברים n והממד d.
   b. זמן הדיווח – במקרה הקשה ביותר יתכן שיש לדווח על כל S. במקרה זה זמן ריצת השאילתה לינארי, למרות שהחיפוש היה יעיל.
2. הגודל – גודל מבנה הנתונים שנבנה כדי ליעל את השאילתה.
3. זמן הבנייה – זמן של עיבוד מקדים שבמהלכו נבנה מבנה הנתונים.

לדוגמא, ניתן לראות את שאילתת החיפוש Search(S,k) כפי שתוארה לעיל כשאילתה בתחום מנוונת. זמן החיפוש הנאיבי יהיה O(n) – פשוט לסרוק את כל איברי הקבוצה S בצורה סדרתית עד שאחד האיברים יקיים את key[x] = k. [CLR1990] מציע שימוש בעץ אדום-שחור (red-black tree) שזמן ביצוע השאילתה הוא O(log n), כאשר זמן ההכנסה וגם זמן המחיקה של איבר הוא O(log n) ; גודלו של העץ הוא O(n).



איור 35: דוגמא לעץ אדום-שחור.

איור 35 נלקח מ- http://en.wikipedia.org/wiki/Red%E2%80%93black_tree. זמן בניית העץ הינו O(n*log n).

נניח שיש לבצע k שאילתות. מתי עדיף להשקיע זמן בעיבוד מקדים יחסית לסריקה הנאיבית?

בסריקה הרגילה העלות של k חיפושים יהיה O(k*n).

במקרה של בניית עץ אדום שחור העלות תהיה מורכבת משני חלקים: בנית עץ פעם אחת וביצוע k שאילתות, לכן העלות תהיה O(n*log n + k*log n).

נפתור את האי-שוויון

$$n*\log n + k*\log n < k*n$$

$$k > \frac{n*\log n}{n-\log n} \qquad (80)$$

עבור n גדול הפעולה מתחילה להיות יעילה אם יש לפחות log n שאילתות לבצע. אם n הוא קטן מאוד, (80) לא תהיה נכונה בגלל המקדמים של הנוסחה שניתן להזניח רק כאשר מדובר במספרים גדולים.

[C1995] בעבודתו מנתח את החסמים של ביצועי החיפוש בתחומים במבנה הכללי בהינתן כמות מוגבלת של הזיכרון m. אם נגדיר ש- n היא כמות הנקודות ו- d הוא ממד המרחב, אז ניתן להגיע לתוצאות הבאות לחסמים הבאים:

פעולה	חסם
גודל מבנה הנתונים	$m = \Omega(n(\log n)^{d-1+\varepsilon}$
זמן ביצוע שאילתה	$(\log n / \log(2m/n))^{d-1}$
זמן ביצוע n הכנסות ו-n שאילתות	$\Omega(n(\log n / \log\log n)^d)$

בעצם ניתן לראות שניתן לשפר את זמן השאילתה על חשבון גודל המבנה ולהיפך.

[AE1997] מתאר את החסמים עבור שאילתות על המישור (d=2) כאשר k היא כמות הנקודות שעונות על השאילתה. שוב מדובר על מבנה הכללי.

סוג שאילתה	גודל	זמן השאילתה
ספירת הנקודות	n	$\log n$
דיווח הנקודות	n	$\log n + k\log^{\varepsilon}(2n/k)$
	$n\log\log n$	$\log n + k\log\log(4n/k)$
	$n\log^2 n$	$\log n + k$
שאילתת קיום	n	$\log n + k$

יש לציין ש-[AE1997] משתמש במודל החישובי של RAM כפי שתוארה ב- [AHU1974].

בשיטה זו תאי זיכרון RAM מחזיקים מספרים שלמים בעלי אורך log n של סיביות. ניתן לעשות על המספרים פעולות חשבוניות פשוטות בזמן קבוע.

6.2 *פעולות על מבני נתונים מרחביים*

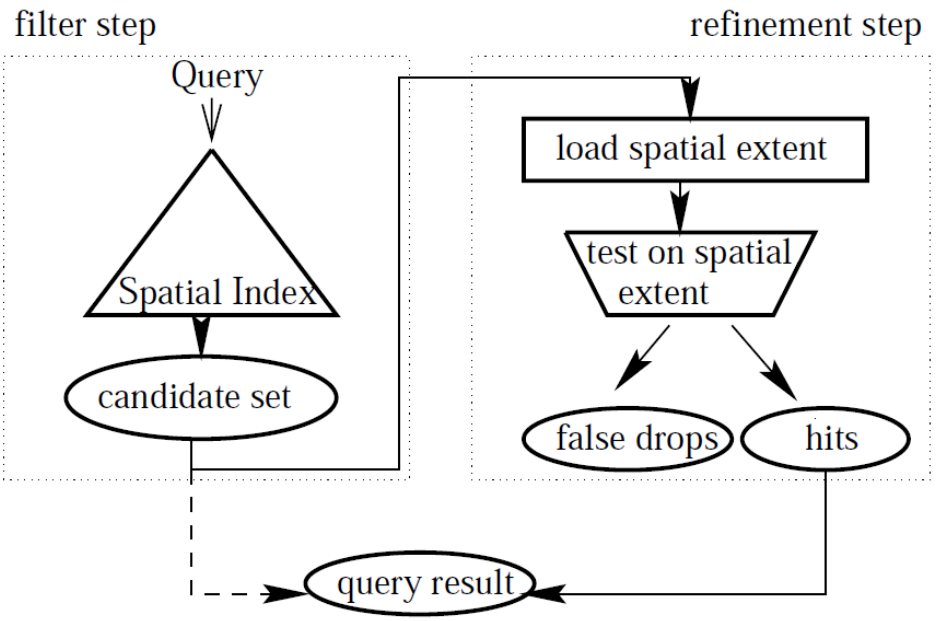
[GG1997] מתאר מספר דרישות ממבנה נתונים מרחבי ומהיישומים שעובדים עם מבנה נתונים כזה:

1. יש לתמוך במספר פעולות כגון הכנסה, מחיקה, שאילתה ולא רק בפעולה אחת.
2. מבנה הנתונים צריך להיות יעיל במרחב (ניצול הזיכרון) ובזמן (זמן ביצוע של שאילתה והכנסה).
3. בחלק מהמקרים יש צורך לתמוך באמצעי זיכרון משני כגון מדיה של אחסון.

ניתן להבחין בין שתי משפחות של מבני הנתונים:

1. PAM - (Point Access Methods) מבנה נתונים שמחזיק נקודות וניתן לבצע עליהם פעולות חיפוש מרחבי.
2. SAM – (Spatial Access Methods) מבנה נתונים שמחזיק ישויות מורכבות יותר, כגון קווים, מצולעים, אליפסות וכו'.

תהליך השליפה מהמבנה המרחבי מתבצע בצורה הבאה:

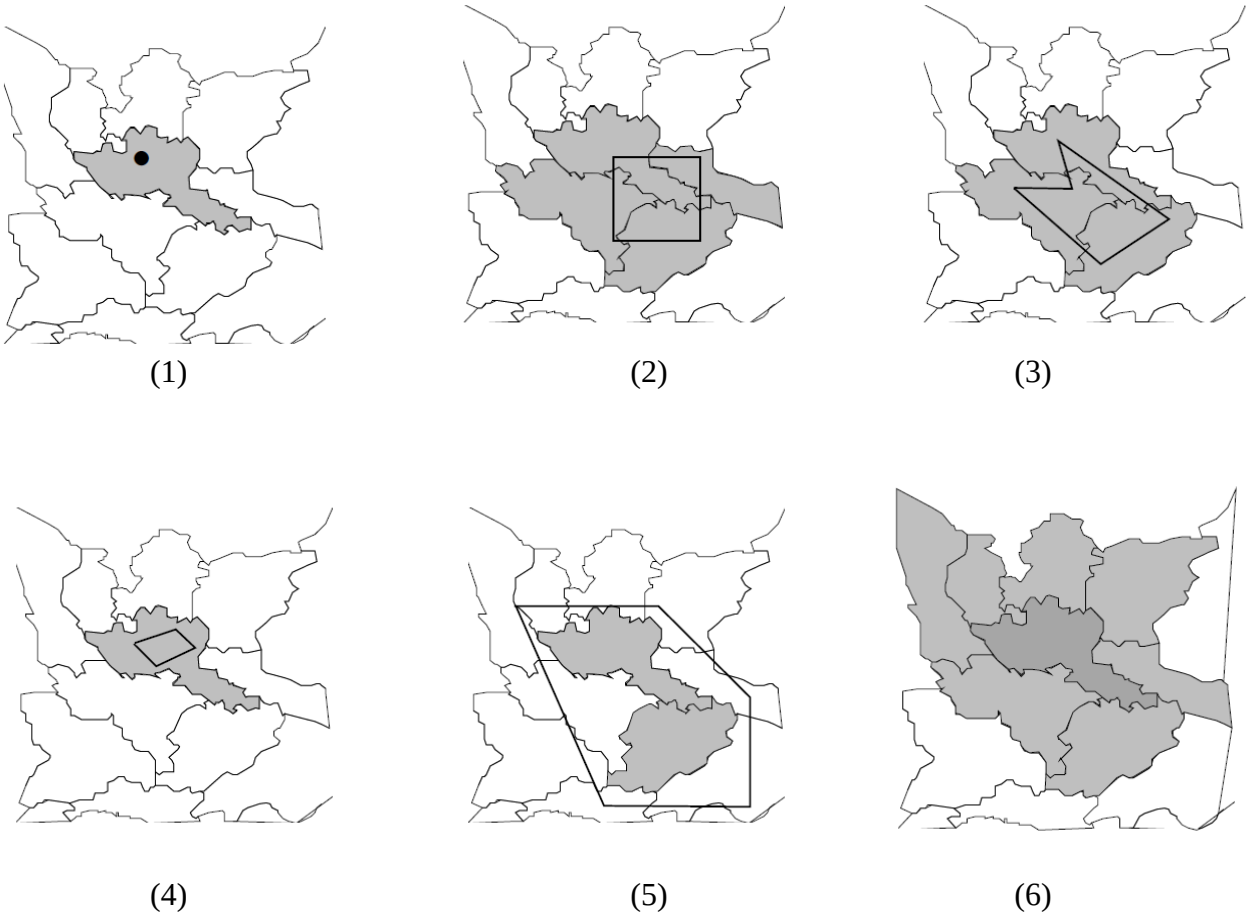


איור 36: תהליך השליפה ממבנה נתונים מרחבי.

איור 36 נלקח מ- [GG1997] .

1. מתקבלת שאילתה מרחבית.
2. מתבצע חיפוש מרחבי בעזרת האינדקסים שמוגדרים במבנה הנתונים ומתקבלת קבוצה של תוצאות אפשריות.
3. טוענים את המבנה המרחבי (הצורה) של אזור השאילתה.
4. בודקים את כל המועמדים משלב 2 כדי לוודא שהם מתאימים לצורה של השאילתה.
5. המועמדים שלא מתאימים נפסלים וכל השאר עוברים כתשובה.

[GG1997] מביא מספר דוגמאות לשאילתות מרחביות:



איור 37: דוגמאות של שאילתות מרחביות .

איור 37 נלקח מ- [GG1997].

(1) שאילתת נקודה (point query). בהינתן נקודה $p \in E^d$ יש למצוא את כל העצמים o בעלי צורה (spatial extent) $o'.G \in E^d$ שכוללים את הנקודה p. ניתן להגדיר את הבעיה בצורה הבאה: יש למצוא את $PQ(p)$, כך ש- $PQ(p) = \{o \mid p \cap o.G = p\}$.

(2) שאילתת חלון (window query). בהינתן תיבה d-ממדית $I^d = [l_1, u_1] \times [l_2, u_2] \times \ldots \times [l_d, u_d]$ יש למצוא את כל העצמים שלפחות אחת הנקודות שלהם משותפת עם I^d, או בצורה פורמלית $WQ(I^d) = \{o \mid I^d \cap o.G \neq 0\}$.

(3) שאילתת חפיפה (intersection/region/overlap query). בהינתן עצם o' בעל צורה $o'.G \in E^d$ יש למצוא את כל העצמים בעלי לפחות נקודה אחת משותפת עם o', או בצורה פורמלית $IQ(o') = \{o \mid o'G \cap o.G\}$.

(4) שאילתת הקפה (enclosure query). בהינתן עצם o' בעל צורה $o'.G \in E^d$ יש למצוא את כל העצמים שמקיפים את o' או בצורה הפורמלית $EQ(o') = \{o \mid o'.G \cap o.G = o'.G\}$

(5) שאילתת הכלה (containment query). בהינתן עצם o' בעל צורה $o'.G \in E^d$ יש למצוא את כל העצמים שמקיפים את o' או בצורה הפורמלית
$$CQ(o') = \{o \mid o'.G \cap o.G = o.G\}$$

(6) שאילתת שכנות (adjacency query). בהינתן עצם o' בעל צורה $o'.G \in E^d$ יש למצוא את כל העצמים שכנים ל- o' או בצורה הפורמלית, אם $o.G^0$ הוא הפנים של $o.G$, אזי $AQ(o') = \{o \mid o'.G \cap o.G \neq 0 \; and \; o'.G^0 \cap o.G^0 \neq 0\}$.

6.3 דוגמאות למבני נתונים לחיפוש

6.3.1 מרחב חד-ממדי (d =1)

פעולות PAM

מציאת העצם המדויק לפי מפתח

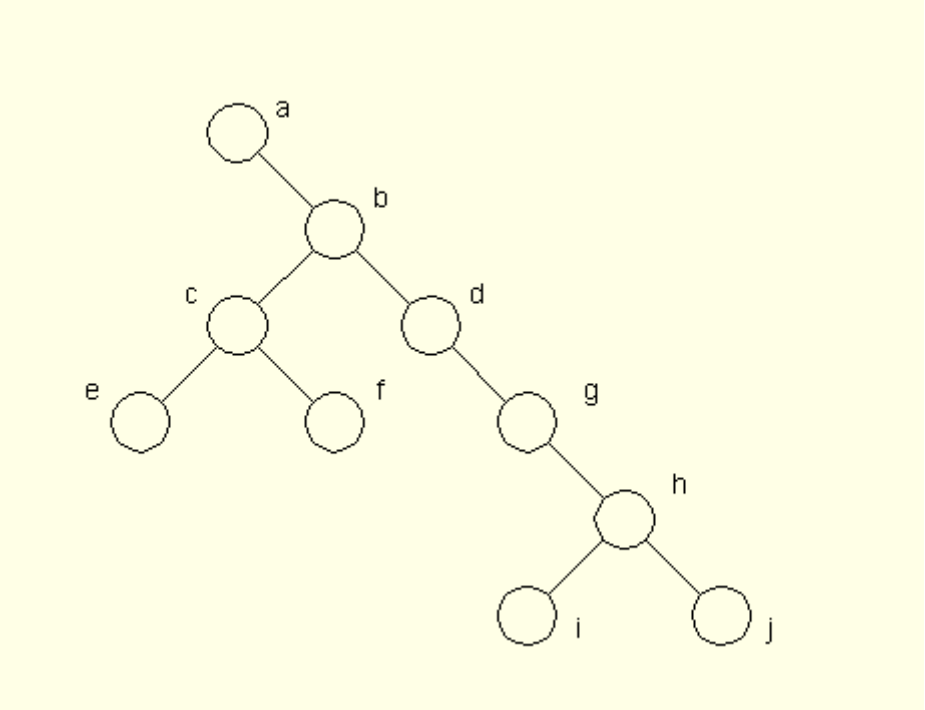
לפעמים בקר אוויר שעוקב אחר תנועת מטוסים נדרש למצוא פרטים אודות מטוס מסוים שנמצא במרחב שלו על פי נתוני זיהוי. אחת הדוגמאות לזיהוי היא אות קריאה של מטוס שאמורה להיות חד חד-ערכית. בעצם, נדרש לבצע חיפוש במרחב חד-מימדי על פי מפתח.

עץ בינרי

פתרון לבעיה זו הוצג לעיל: עץ חיפוש אדום-שחור (RB-tree). באופן כללי עצי חיפוש בינריים הם מבני נתונים היררכיים בעלי התכונות הבאות [CLR1990]:

1. לכל צומת יש לכל היותר שני בנים.
2. אם x הוא צומת בעץ ו- y הוא צומת בתת-עץ השמאלי של x, אז $key[y] < key[x]$, לעומת זאת אם y הוא צומת בתת-עץ הימני של x, אז $key[y] > key[x]$.

אוסף תכונות זה מאפשר לבצע סקירה של הצמתים בצורה ממוינת בעזרת אלגוריתם רקורסיבי פשוט (inorder tree walk). בעצם, מבנה נתונים מסוג זה מחזיק את כל הנתונים בצורה ממוינת. אם מכניסים או מוציאים איבר ממבנה הנתונים, העץ שומר על המיון של כל האיברים. המגבלה של מבנה נתונים זה היא ההיררכיות שלו: תמיד מתחילים מהשורש וכדי להגיע לצומת כלשהו על מנת לבצע עליו את הפעולות, נדרש לעבור על כל המסלול בעץ מהשורש עד הצומת הנדרש.

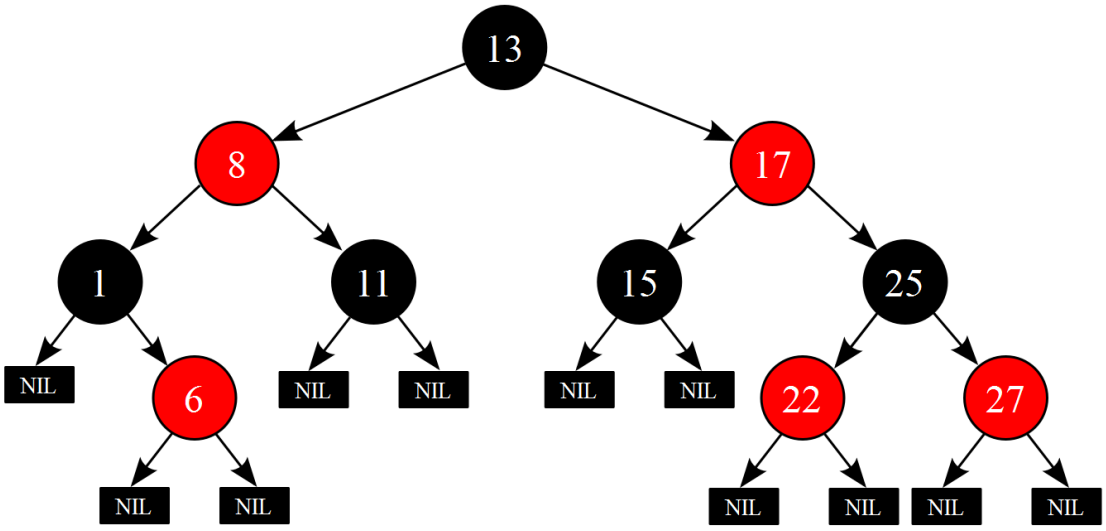


איור 38: דוגמא לעץ בינרי.

איור 38 נלקח מ-

http://pages.cpsc.ucalgary.ca/~jacobs/Courses/cpsc331/W12/tutorials/tutorial10.html

במקרה הגרוע ביותר, יתכן עץ בינרי שהוא בעצם רשימה משורשרת, במקרה של האיור העץ יהיה מורכב מהצמתים (a,b,d,g,h,j). במקרה כזה זמן ההגעה לצומת אחרון (j במקרה זה) יהיה $c \times n = O(n)$ וגם התוחלת של זמן ההגעה לצומת כלשהו יהיה $\frac{c \times n}{2} = O(n)$! על מנת שהעץ יהיה יעיל הוא צריך להיות מאוזן: כלומר, לכל צומת, ההבדל בגובה בין שני התת-עצים של הצומת לא יעלה על כמות מודגרת, למשל 1.



איור 39: דוגמא לעץ מאוזן.

איור 39 נלקח מ- http://en.wikipedia.org/wiki/Red%E2%80%93black_tree. עץ RB הוא עץ חיפוש בינרי בעל התכונות הבאות [CLR1990]:

1. לכל צומת יש שדה צבע (אדום או שחור).
2. כל עלה הוא שחור וערכו הוא nil (מצביע ריק).
3. לכל צומת אדום יש שני בנים שחורים.
4. לכל צומת נתון, בכל מסלול פשוט בין צומת זה לכל עלה צאצא שלו נמצאת אותה כמות של צמתים שחורים.

מהתכונות האלו נובעות התוצאות הבאות:

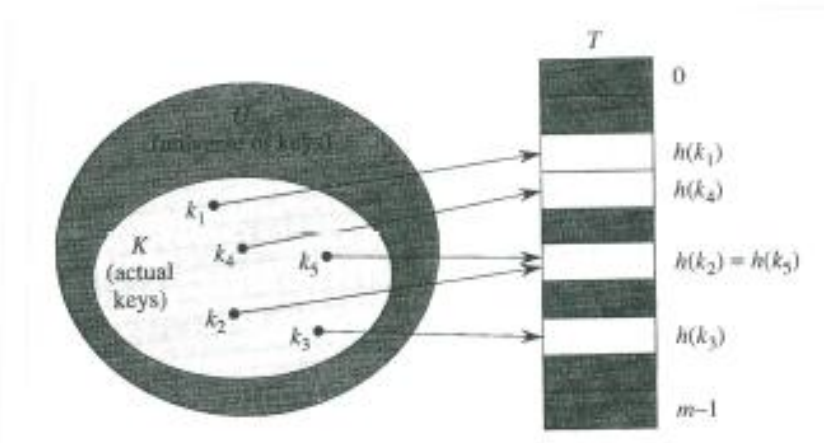
1. אם בעץ יש n צמתים, גובה העץ הוא לכל היותר $2\log(n+1)$.
2. עץ RB הוא כמעט מאוזן.

כפי שצוין קודם הביצועים של העץ הם [CLR1990]:

פעולה	עלות
גודל מבנה הנתונים	n
זמן ביצוע שאילתה	$O(\log n)$
זמן ביצוע הכנסה/מחיקה של איבר בודד	$O(\log n)$
זמן בנייה של המבנה כולו	$O(n\log n)$

טבלת גיבוב

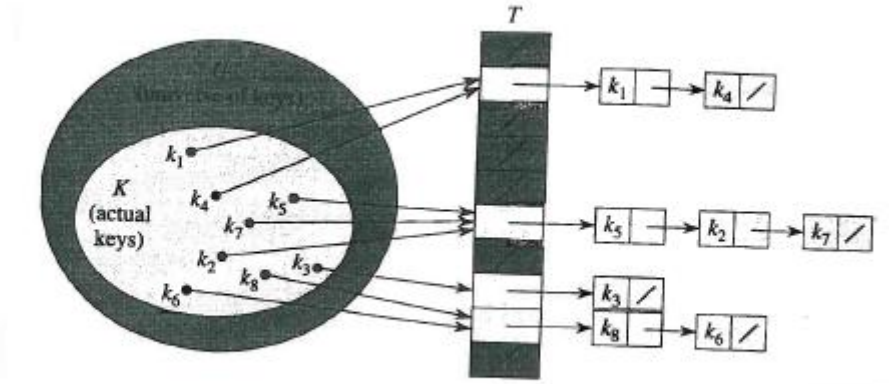
אם מספר העצמים שנדרש לטפל בהם קטן יחסית לכל העצמים הפוטנציאליים, אז אפשר להשתמש בשיטה זו. לכל עצם מקצים רשומה וגם מחשבים את הכתובת שלו בעזרת פונקציה פסידו-מקרית על המפתח של העצם ($key[x]$) h. כאשר רוצים לגשת לעצם שוב מחשבים את כתובת הרשומה וניגשים אל העצם.



איור 40: דוגמא לשימוש בגיבוב.

איור 40 נלקח מ- [CLR1990].

שיטה זו יעילה מאוד כאשר כמות האיברים שיש לשמור ב- m רשומות הוא קטן יחסית לכמות האיברים הפוטנציאליים n. ברור שאם כמות האיברים גדולה יותר מ- m, אז תחול התנגשות. קיים מספר טכניקות להתמודד עם ההתנגשות למשל לשמור את כל האיברים בעלי אותה כתובת ברשימה משורשרת.



איור 41: דוגמא להתמודדות עם ההתנגשויות בגיבוב[CLR1990].

איור 41 נלקח מ- [CLR1990]

הבעיה היא שיש לעבור בצורה סדרתית על כל הרשימה על מנת להגיע לעצם הנדרש.

הביצועים של טבלת הגיבוב הם [CLR1990]:

פעולה	עלות
גודל מבנה הנתונים	m
זמן ביצוע שאילתה	$\Theta(1+\alpha)$
זמן ביצוע הכנסה/מחיקה	$\Theta(1+\alpha)$
זמן בנייה של המבנה כולו	$m\Theta(1+\alpha)$

כאשר $\alpha = n/m$ הוא גורם העומס (load factor). אפשר בקלות לראות שיעילות הפעולות באה על חשבון יעילות המקום. נקודת האיזון היא $m = \sqrt{n}$, במקרה כזה גם הגודל יהיה $\Theta(\sqrt{n})$ וגם הפעולה תיקח $\Theta(\sqrt{n})$. בהינתן פונקציה h יעילה, ברור שמבנה של גיבוב עדיף על המבנה של עץ חיפוש (בחיפוש העצם המדויק): באותה עלות של מקום, פעולות על הקבוצה הדינמית משמעותית יותר זולות. היתרון של עץ חיפוש מאוזן היא חסם עליון על זמן החיפוש, כאשר מבנה הנתונים של גיבוב מבטיח רק את הזמן הממוצע.

מציאת העצם בתחום מסוים

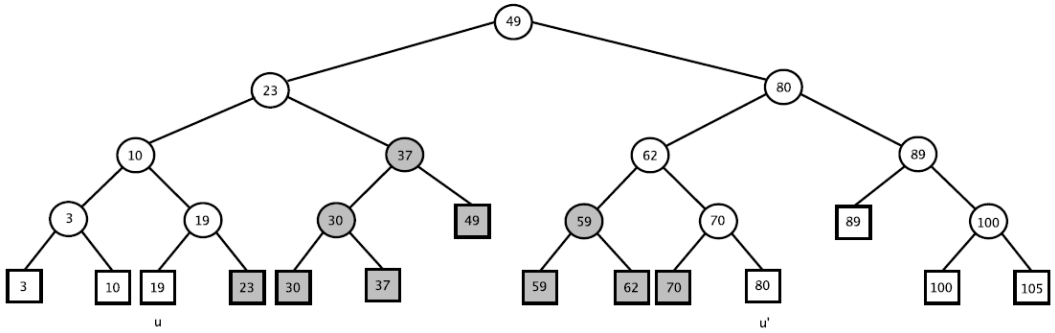
ניתן להרחיב את בעיית מציאת מטוס במרחב לבעיה של סימון כל המטוסים בעלי קוד זיהוי שנמצא בתחום מסוים. לדוגמא, נדרש למצוא את כל המטוסים ששייכים לטייסת מסוימת.

בעיה זו היא בעצם שאילתת חלון (window query) שהוגדרה לעיל : בהינתן $I^d = [l_1,u_1]\times[l_2,u_2]\times...\times[l_d,u_d]$, יש למצוא את $WQ(I^d) = \{o \mid I^d \cap o.G \neq 0\}$.

עץ חיפוש

[BKOS2000] מתאר את הפתרון עבור d=1.

נגדיר את קבוצת העצמים $P := \{p_1, p_2, ..., p_n\}$. נבנה עץ T כך שהעלים יכילו את המפתחות של העצמים P והצמתים הפנימיים יכילו את ערכי הפיצול שנועדו לעזור לחיפוש.



איור 42: תאור עץ חיפוש תחומים d=1 .

איור 42 נלקח מ- [BKOS2000]. נגדיר את ערך הפיצול שנמצא בצומת v כ- x_v. כאשר המפתחות של כל העצמים בתת-עץ השמאלי של v קטנים מ- x_v ולהיפך: המפתחות של כל העצמים בתת-עץ ימני של v גדולים מ- x_v. פעולת החיפוש עובדת בצורה הבאה: נניח נתונה שאילתה על הקטע $[u:u']$. אם נחפש את העצם u, החיפוש יסתיים בעלה μ, ובהתאם החיפוש אחרי העצם u' יסתיים ב- μ'. פעולת החיפוש תחזיר את כל העלים שנמצאים בין $[\mu:\mu']$.

למשל, אם השאילתה הייתה למצוא את העצמים בין [20:75], החיפוש יסתיים בצמתים 19 ו- 80. השאילתה תחזיר את הערכים: [23,30,37,59,62,70]. כדאי לציין שקבוצת הערכים הפוטנציאליים הייתה יותר גדולה ממה שדווח בפועל: היא כללה גם את 19 ו- 80. אבל הערכים אלו נפלו על בדיקה של צורת החלון.

ביצועיו של עץ החיפוש בתחומים עבור d=1 הם [BKOS2000]:

פעולה	עלות
גודל מבנה הנתונים	$2n = O(n)$
זמן ביצוע שאילתה	$O(\log n)$
זמן ביצוע הכנסה/מחיקה	$O(\log n)$
זמן בנייה של המבנה כולו	$O(n \log n)$

פעולות SAM

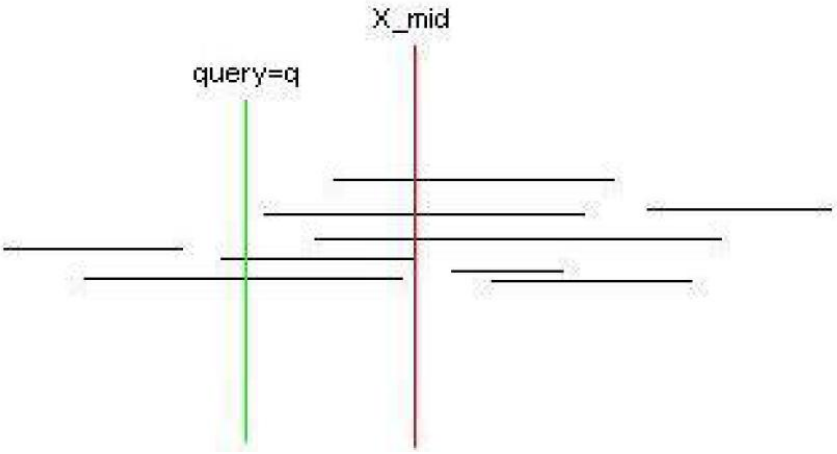
מציאת רווחים

מנתח תמונה אווירית עשוי להתמודד עם בעיה נוספת: מי הם המטוסים שהיו בעקיבה בחלון זמן מסוים. לדוגמא, אם ידוע שהייתה חדירה לתחום האווירי בזמן מסוים יכול להיות מאוד חשוב לדעת אילו מטוסים היו בזירה בזמן הסמוך לזמן החדירה.

עץ רווחים

בפעולות SAM מטפלים בעצמים שהם יותר מורכבים מנקודה. דוגמא לעצם כזה ב- d=1 הוא רווח סגור על הקו (closed interval). אחת הבעיות שיכולות להיות היא שאילתת דקירה: אילו רווחים מכילים את הנקודה q. שאילתה זו היא בעצם שאילתת הכלה, בהינתן עצם o' בעל צורה $o'.G \in E^d$ יש למצוא את $CQ(o') = \{o \mid o'.G \cap o.G = o.G\}$.

[BKOS2000] מתאר את הפתרון לבעיה זו בצורה הבאה. נגדיר את $I = \{[x_1, x'_1],[x_2, x'_2],...,[x_n, x'_n]\}$. מבנה הנתונים המוצע הוא *interval tree* עץ רווחים.



איור 43: תאור הבעיה.

איור 43 נלקח מ- [BKOS2000]. הנקודה x_{mid} מתארת את נקודת החציון של 2n נקודות הקצה של הרווחים. מבנה הנתונים המוצע מוגדר בצורה רקורסיבית:

אם $I = \phi$, אז העץ הוא עלה

אחרת,

נגדיר שלוש קבוצות:

$$I_{left} := \{[x_j : x'_j] \in I : x'_j < x_{mid}\}$$
$$I_{mid} := \{[x_j : x'_j] \in I : x_j < x_{mid} < x'_j\}$$
$$I_{right} := \{[x_j : x'_j] \in I : x_j > x_{mid}\}$$

אז העץ יכיל צומת שורש v שיכיל ערך x_{mid} ,

התת-עץ השמאלי של v יכיל את עץ הרווחים על I_{left}, והתת-עץ הימני, בהתאם, יכיל את I_{right}.

כמו כן, נשייך לכל צומת v את מבנה שיכיל את I_{mid}. העץ שמתקבל הוא סוג של עץ בינרי כאשר כל אחד מהצמתים מחזיק מבנה מורכב. השאילתה תתבצע בצורה הבאה: כאשר מתקבלת נקודת דקירה q_x, בודקים האם היא נמצאת ימנית או שמאלית יחסית ל- x_{mid} וממשיכים לבדוק את התת-עץ המתאים. ההיגיון מאחרי זה הוא שאם $q_x < x_{mid}$, אז כל הרווחים ששיכים ל-I_{right} נמצאים ימנית ל- x_{mid} ולכן גם ימנית ל- q_x. אומנם יש לבדוק גם את קבוצת I_{mid}, אם הרווחים שנמצאים שם מכילים את נקודת הדקירה. משימה זו קלה יותר כיוון שידוע שכל אחד מהרווחים שנמצאים בתוך ה- I_{mid} מכיל את x_{mid} לפי הגדרת I_{mid}. לכן אם הרווחים מאוחסנים בצורה ממוינת, ניתן לסקור אותם בצורה סדרתית משמאל לימין וברגע שיש רווח שלא מכיל את q_x, המשמעות היא ש- q_x נמצא שמאלית משאר הרווחים ולכן אין צורך להמשיך בבדיקה. כמובן, את הבדיקה יש לעשות פעמיים: פעם אחת משמאל ופעם אחת מימין. אי לכך ייסרקו רק הרווחים שידווחו, ולכן סיבוכיות הפעולה הזו היא $O(k)$.

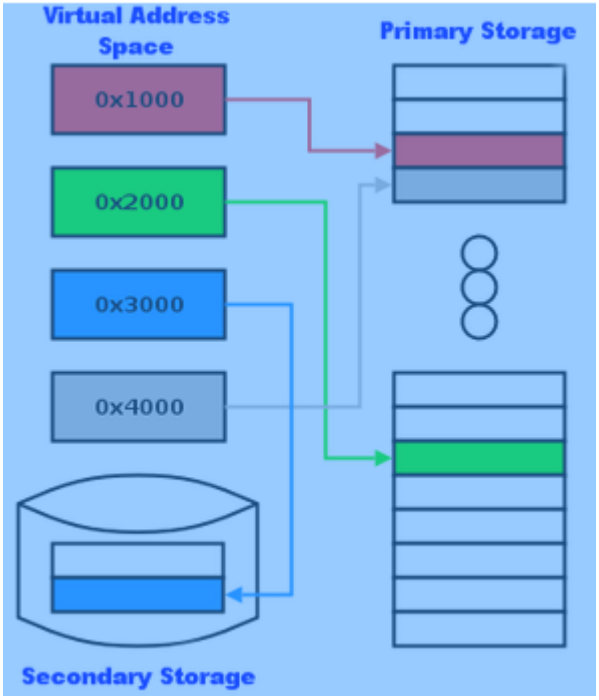
[BKOS2000] מראה שהעומק המקסימלי של העץ הוא $O(\log n)$.

הביצועים של עץ המקטעים d=1 הם [BKOS2000]:

פעולה	עלות
גודל מבנה הנתונים	$O(n)$
זמן ביצוע שאילתה	$O(\log n + k)$
זמן ביצוע הכנסה/מחיקה	$O(\log n)$
זמן בנייה של המבנה כולו	$O(n \log n)$

זיכרון משני.

בדוגמאות שתוארו קודם, כל הנתונים מוחזקים בזיכרון ראשי, אומנם גודל זיכרון זה הוא מוגבל ואם כמות הנתונים שיש לטפל בהם גדולה מאוד יש להשתמש בזיכרון משני לאחסון הנתונים. הדוגמא הטובה ביותר לכך היא בסיס נתונים שבדרך כלל מחזיק את כל הנתונים שלו על הדיסק. בדרך כלל, גישה לנתונים שעל הדיסק נעשית בעזרת מערכת ניהול קבצים של מערכת ההפעלה או של בסיס הנתונים עצמו. יחידת בסיס שאליה נדרש להגיע היא בדרך כלל רשומה (record) שיכולה להיות בגודל משתנה. לעומת זאת יחידת אחסון הבסיסית היא עמוד (page) שגודלו קבוע בכל מערכת ההפעלה, כך יתכן בדרך כלל שבעמוד אחד תהיה יותר מרשומה אחת. איור 44 נלקח מ- http://blog.csdn.net/tangbomaozero/article/details/936680

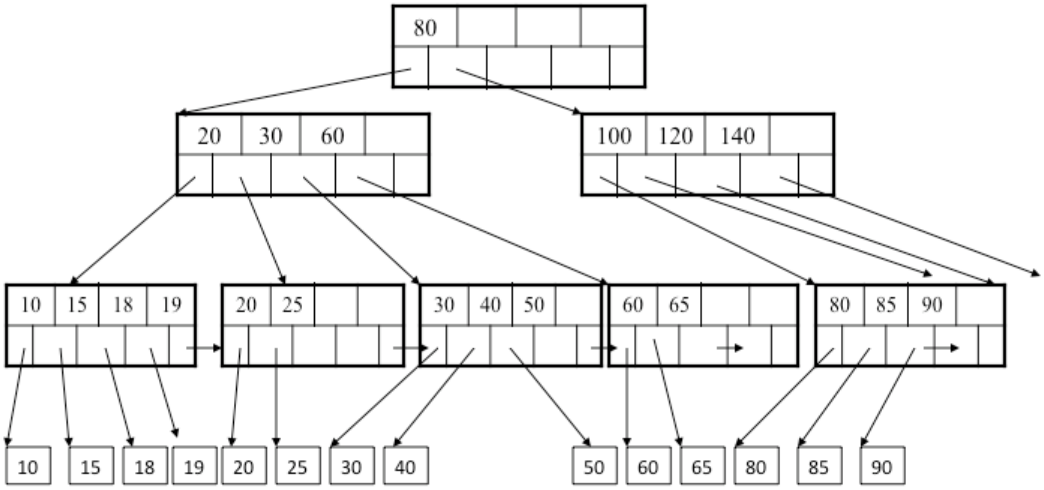


איור 44: דוגמא לשימוש בזיכרון משני.

כאשר נדרש לגשת לרשומה יש צורך לטעון לזיכרון הראשי את העמוד שמכיל את הרשומה הנדרשת. גישה לזיכרון המשני היא בדרך כלל פעולה מאוד יקרה כי היא מצריכה פעולה מכנית. לכן משתמשים במנגנוני אינדוקס אשר אמורים להקטין משמעותית את כמות הגישות לזיכרון המשני כפי שמתואר ב- [RG2003].

עץ B (B-tree)/ עץ B+ (B+-tree)

אחת הגישות של האינדוקס היא שימוש בעצי B+, למשל ב- IBM DB2, Informix , Microsoft SQL Server ו- Oracle 8.



איור 45: דוגמא לעץ B/B+ tree.

איור 45 נלקח מ- http://blog.csdn.net/p424671075/article/details/6907598.

ההגדרה של עץ B היא הבאה [CLR1990]:

1. כל צומת x מכיל את השדות הבאים:
   a. $n[x]$ – מספר המפתחות שכרגע בשימוש, שמאוחסנים בצומת x.
   b. המפתחות מסודרים בצורה מונוטונית לא יורדת
   $$key_1[x] \le key_2[x] \le ... \le key_n[x]$$
2. לכל צומת פנימי (לא עלה) נמצאים $n[x]+1$ מצביעים לצמתים הבנים: $c_i[x]$
3. הערכים של $key_i[x]$ מגדירים מהם הערכים שנמצאים בתת-עץ הבן. הכלל אומר שהערכים של מפתחות שנמצאים בבן צריכים להיות קטנים מ$key_i[x]$ של צומת האב, אבל גדולים מ- $key_{i-1}[x]$. לדוגמא באיור 45: המפתח הראשון בצומת שורש $root$ הוא 80, $c_i[root]$ מצביע על הצומת $node$ שמכיל ערכים הבאים: 20,30,60. כולם קטנים מ-80.
4. כל העלים נמצאים באותו עומק, שהוא בעצם הגובה של העץ h.
5. החסמים לכמות המפתחות שכל צומת יכול להכיל מוגדרים כפונקציות של מספר שלם $t \ge 2$:
   a. כל צומת חוץ מהשורש צריך להכיל לפחות $t-1$ מפתחות.
   b. כל צומת יכול להכיל לכל היותר $2t-1$ מפתחות.

בנייה של העץ B בעל ערך t מסוים נעשית בצורה הבאה:
בהתחלה בונים את השורש. מכניסים ערכים לצומת השורש בצורה ממוינת. ברגע שכמות האיברים בצומת עולה על t, מבצעים פעולת הפיצול (split) וחצי מהאיברים הגדולים מעבירים לצומת חדש. לאחר מכן בונים צומת חדש שלתוכו מכניסים את הערך המינימלי של הצומת השני. הצומת החדש כעת יהיה השורש של העץ. לאחר מכן ממשיכים למלא את הצמתים תוך כדי ביצוע פיצול כאשר צמתים מתמלאים ומחזיקים יותר מ t איברים. בשלב שבו מתמלא צומת השורש, הפיצול שלו יגרום ליצירת שורש חדש והגדלת עומק של העץ כולו.
החיפוש בעץ-B דומה לחיפוש בעץ בינרי בהבדל אחד: בכל צומת, כדי לקבל החלטה לאיזה תת-עץ להמשיך, נדרש לעבור על n אפשרויות ולא על 2.

פעולה	עלות
גודל מבנה הנתונים	$O(n)$
זמן ביצוע שאילתה	$O(t \log_t n)$
זמן ביצוע הכנסה/מחיקה	$O(t \log_t n)$

בבסיסי נתונים משתמשים לעיתים ב-bulk insert, כלומר הכנסת נתונים בחבילה. הרעיון הוא לחסוך בפעולות הפיצול שהן יקרות יחסית. בשיטה זו קודם ממלאים את כל העלים ואז ממשיכים לבנות את צמתי אב.

עץ-B+ הוא מודיפיקציה של עץ B. ההבדל הוא שבעלים של עץ-B+ נמצאים מצביעים (pointers) למידע המאוחסן ולא מידע עצמו, כלומר, מבנה נתונים זה מהווה אינדקס של נתונים שנמצאים במקום אחר [RG2003]. לעומת זאת, עץ B מחזיק את המידע עצמו בתוך העלים שלו.

6.3.2 מרחב דו-ממדי (d = 2)

פעולות PAM

מציאת נקודה במלבן.

אחד הצרכים שקיימים הוא למצוא את כל המטוסים שנמצאים באזור מסוים. לדוגמא, במקרה של נחיתת חרום רוצים לשנות נתיבים לכל המטוסים שנמצאים בסביבה.

עץ-Kd (Kd-tree).

בעיית המציאה של הנקודות שנמצאות בתוך המלבן של השאילתה היא ההרחבה של מציאת נקודה בתוך הקטע, או במלים אחרות שאילתת חלון. חשיבותה של בעיה זו היא מאוד גדולה: פתרונה מאפשר חיפוש עצמים גאוגרפיים, למשל למצוא את כל שדות התעופה או לחלופין סוללות תק״א הנמצאות באזור של המטוס שטס . פקח מוניות יכול לאתר את כל המוניות שנמצאות באזור הלקוח שביקש שירות ועוד דוגמאות נוספות. ברור שבמקרה של יותר מממד אחד אי אפשר להיעזר במבני הנתונים שבנויים על מיון בממד אחד. יש צורך להציע מבנה נתונים שממוין בו-זמנית לפי שתי הקואורדינטות.

אחד הפתרונות המקובלים הוא שימוש ב- kd-tree [BKOS2000].

הבעיה מוגדרת בצורה הבאה: נתונה קבוצת נקודות

$P := \{p_1(x_1, y_1), p_2(x_2, y_2), ..., p_n(x_n, y_n)\}$, נדרש למצוא את כל הנקודות השייכות ל-P

ונמצאות בתוך מלבן $Q := [x : x'] \times [y : y']$. הבעיה היא שקל אומנם למיין את הערכים לפי קואורדינטה מסוימת וכך לחלק את המרחב לתת-מרחבים אבל חלוקה זו בכלל לא תקפה לגבי הקואורדינטה השנייה ואז גם החלוקה לא תהיה תקפה. הפתרון המוצע ב-kd-tree הוא לבצע בכל שלב חלוקת המרחב לפי קואורדינטה אחרת, כלומר, יש עץ שכל הרמות הזוגיות שלו מנהלות עץ חיפוש לפי קואורדינטת x, וכל הרמות האי-זוגיות לפי קואורדינטת y.

האלגוריתם של בניית העץ הוא רקורסיבי :

בכל שלב מקבלים את רשימת הנקודות P ואת העומק הנוכחי של העץ *depth*.

אם P כולל רק נקודה אחת, יש להחזיר עלה שמכיל את הנקודה,

אחרת:

אם העומק הוא זוגי (עבודה על קואורדינטת x):

מוצאים נקודת חציון של כל קואורדינטות x של P.

מחלקים את כל הנקודות בעזרת קו אנכי l שעובר דרך נקודת החציון ואז מקבלים שתי תת קבוצות P_1 משמאל לקו ו- P_2 לימינו.

אם העומק הוא אי-זוגי (עבודה על קואורדינטת y):

מוצאים נקודת חציון של כל קואורדינטות y של P.

מחלקים את כל הנקודות בעזרת קו אופקי l שעובר דרך נקודת החציון ואז מקבלים שתי תת-קבוצות P_1 מתחת לקו ו- P_2 מעליו.

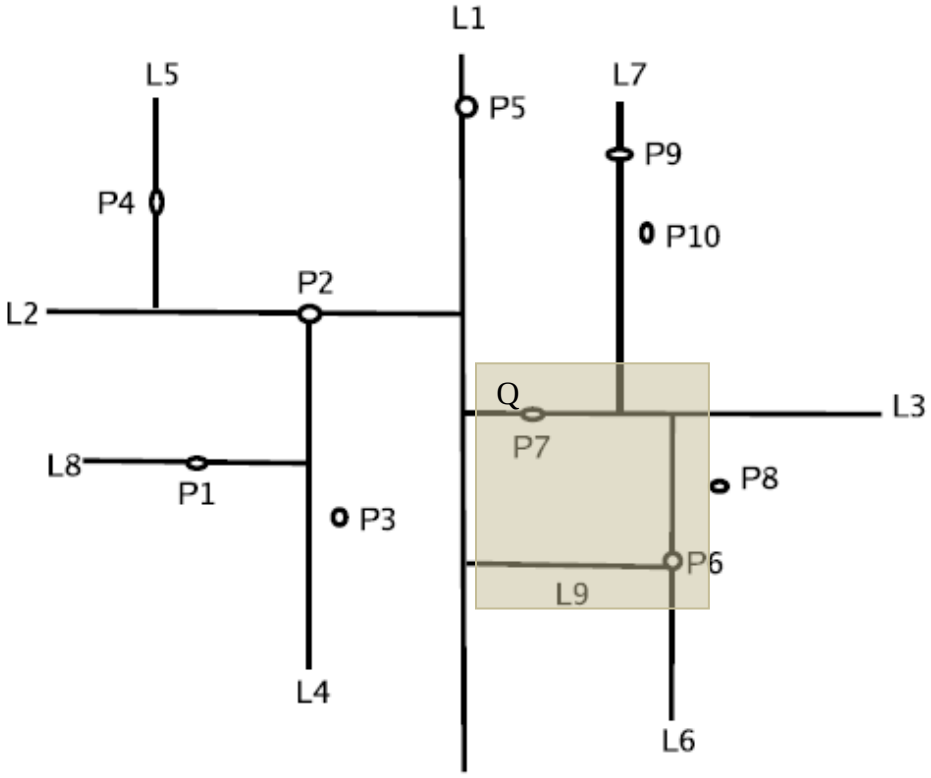
בונים תת עץ $v_{left} \leftarrow BuildKDTree(P_1, depth+1)$

בונים תת עץ $v_{right} \leftarrow BuildKDTree(P_2, depth+1)$

בונים צומת v שמכיל את הקו l , v_{left} יהיה התת-עץ השמאלי שלו ו- v_{right} יהיה התת-עץ הימני שלו.

מחזירים את הצומת v.

מתחילים את האלגוריתם ע"י הפעלה של $KDTree \leftarrow BuildKDTree(P, 0)$.

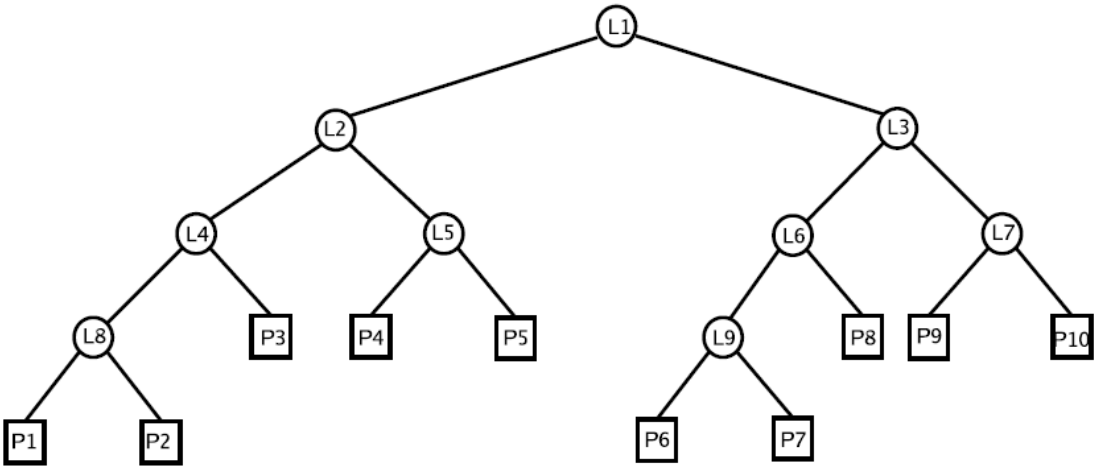


איור 46: תאור של המודל הגאומטרי של kd-tree.

איור 46 מבוסס על [BKOS2000].

באיור אפשר לראות שהתקבל P בעל 10 נקודות. בשלב ראשון, האלגוריתם עבד על הקואורדינטה x. נבחרה נקודה p_5 ,דרכה עבר קן l_1 , שחילק את הנקודות לשתי תת-קבוצות: P_1 משמאל לקו ו- P_2 לימינו. לאחר מכן האלגוריתם הופעל על P_1 , נבחרה נקודה p_2 ודרכה עבר הקו האופקי l_2 , וגם על P_2 , שם נבחרה נקודה p_7 ודרכה עבר קו אופקי l_2 וכך הלאה.

מבנה נתונים זה ייראה בצורה הבאה:



איור 47: הצגת kd-tree בצורת עץ.

איור 47 נלקח מ- [BKOS2000]. כל עלה מחזיק את הנקודות של P ושאר הצמתים את הקווים המתאימים לאיור למעלה, למשל השורש של העץ מחזיק את הקו l_1, הבן השמאלי את l_2 וימני את l_3. אפשר להגדיר מושג *region(v)* שמתאר את האזור שמוגדר ע"י צומת *v*. למשל, עבור השורש *root* , *region(root)* מתאר את כל המישור. לעומת זאת עבור הצומת *v* שמחזיק את l_9, *region(v)* מתאר את האזור שמוגבל ע"י הקווים l_1, l_3, l_6, l_9. אפשר לראות באיור למעלה שהנקודות p_6 ו- p_7 נמצאות באזור זה, לכן נקודות אלו נמצאות בעלים שהם הצאצאים של צומת *v*. אם כך, ניתן להגיד שהתשובה לשאילתה מלבנית Q, יש למצוא את צומת *v* הגבוה ביותר כך שכולו נמצא בתוך Q. ולכן תהליך החיפוש יהיה לעבור על העץ דרך הצמתים כך ש- *region*() שלהם יחתוך את המלבן Q.

בדוגמא הזאת אף אזור לא נמצא כולו בתוך מלבן השאילתה Q, אבל האזורים הבאים כן חותכים אותו: l_1, l_3, l_6, l_7, l_9. אומנם אף אחת מהנקודות של $region(l_7)$ לא נמצאת בתוך Q, לעומת זאת הנקודות p_6 ו- p_7 של האזור $region(l_9)$ כן בפנים, לכן התשובה תהיה p_6 ו- p_7 .

הביצועים של העץ kd-tree (d=2) הם[BKOS2000]:

פעולה	עלות
גודל מבנה הנתונים	$O(n)$
זמן ביצוע שאילתה	$O(\sqrt{n}+k)$
זמן ביצוע הכנסה/מחיקה	$O(\log n)$
זמן בנייה של המבנה כולו	$O(n\log n)$

יש לציין שזמן ביצוע השאילתה במקרה זה הוא לא לוגריתמי.

פעולות SAM

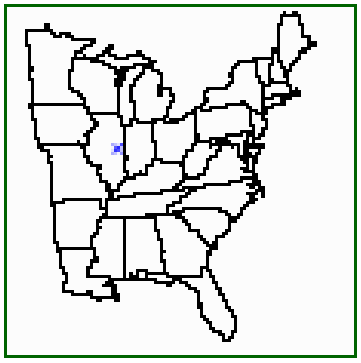
מציאת מצולע מסביב לנקודה נתונה.

לפעמים הבעיה היא הפוכה: ישנו מטוס ונדרש להבין באיזה אזור הוא נמצא. לדוגמא, נדרש להבין לאיזה אזור פיקוח/בקרה הוא שייך. סיבה נוספת יכולה להיות הצורך להתריע אם המטוס נכנס לאזור האסור.

עץ רביעים (Quadtree).

שאילתה כזו היא מסוג PointQuery.

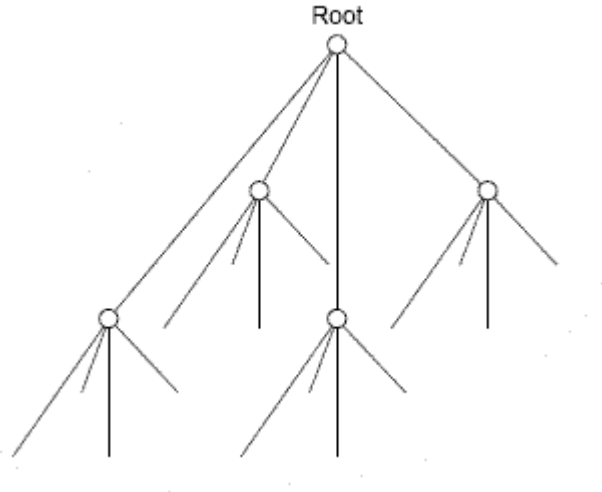
נדרש למצוא את כל המצולעים שמכילים את נקודת p.



איור 48: דוגמא לשאילתה [H2007].

איור 48 נלקח מ- [H2007]. אחד הפתרונות המקובלים הוא שימוש בעץ רביעים [H2007].

עץ רביעים הוא עץ שבו לכל צומת פנימי יש בדיוק 4 בנים [BKOS2000].

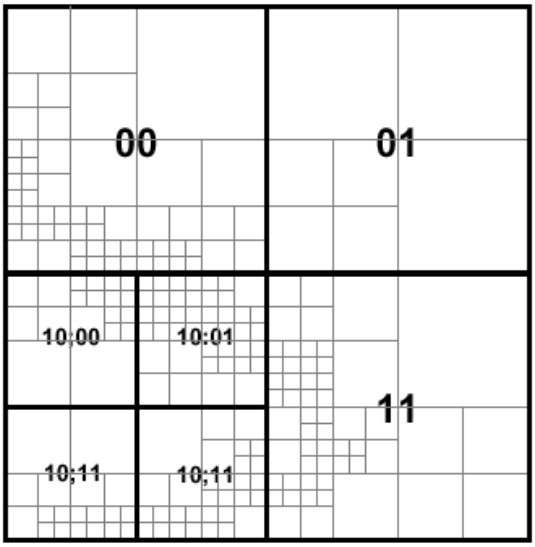


איור 49: דוגמא לעץ רביעים.

איור 49 נלקח מ http://jcsites.juniata.edu/faculty/rnodes/smui/parex.html.

אפשר לקרוא לבנים בשמות: צפון-מזרח NE, דרום-מזרח SE, צפון-מערב NW ו- דרום-מערב SW. הרעיון הגאומטרי אומר שצומת השורש מייצג ריבוע בעל אורך s שמכיל את כל אזור העניין.

ניתן לחלק את הצומת v ל- 4 ריבועים זהים, כל אחד בעל צלע באורך $\frac{s}{2}$. כאשר כל אחד מהריבועים החדשים הוא בעצם הבן של הצומת v. במידת הצורך ניתן לחלק גם את הבנים ל- 4 תת-ריבועים כל אחד.



איור 50: דוגמא לחלוקה לתת-ריבועים.

איור 50 נלקח מ-
http://geodata.ethz.ch/geovite/tutorials/L2GeodataStructuresAndDataModels/en/html/unit_u2Raster.html.

ההגדרה של Quadtree עבור קבוצת נקודות P בתוך הריבוע σ תראה כך: נניח שריבוע σ מוגדר בצורה הבאה: $\sigma := [x_\sigma : x'_\sigma] \times [y_\sigma : y'_\sigma]$

- אם P כולל רק נקודה אחת או קבוצה ריקה של נקודות, אז העץ יכלול צומת אחד בלבד שיכלול את P ואת σ.
- אם P כולל שתי נקודות או יותר, אז נגדיר את $\sigma_{NE}, \sigma_{NW}, \sigma_{SE}, \sigma_{SW}$ כארבעת הרביעים של σ. נגדיר $x_{mid} := \frac{x_\sigma + x'_\sigma}{2}; y_{mid} := \frac{y_\sigma + y'_\sigma}{2}$ ואז נגדיר תת קבוצות של P בצורה הבאה:
  - $P_{NE} := \{p \in P : p_x > x_{mid} \text{ and } p_y > y_{mid}\}$
  - $P_{NW} := \{p \in P : p_x \le x_{mid} \text{ and } p_y > y_{mid}\}$
  - $P_{SE} := \{p \in P : p_x > x_{mid} \text{ and } p_y \le y_{mid}\}$
  - $P_{SW} := \{p \in P : p_x \le x_{mid} \text{ and } p_y \le y_{mid}\}$

לעץ יהיה בשלב זה שורש שמכיל את הריבוע σ וארבעת הבנים NE, SE, NW ו- SW שמכילים את הרביעים המתאימים ואת הנקודות המתאימות P_{NE}, P_{SE}, P_{NW} ו- P_{SW}.
בניית העץ תתבצע בחלוקה רקורסיבית של הרביעים עד שבכל אחד מהרביעים תישארנה, לכל היותר, שתי נקודות.

[BKOS2000] מוכיח שגובה העץ המקסימלי יכול להיות $O(\log(\frac{s}{c}) + \frac{3}{2})$.

לפי [BKOS2000] הביצועים של Quadtree הם הבאים:

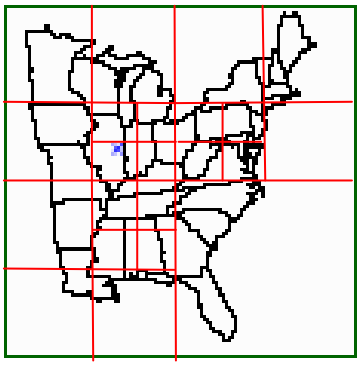
פעולה	עלות
גודל מבנה הנתונים	$O((d+1)n)$
זמן ביצוע שאילתה	$O(d)$
זמן ביצוע הכנסה/מחיקה	$O(d)$
זמן בנייה של המבנה כולו	$O((d+1)n)$

כאשר d הוא העומק המקסימלי של העץ.

העץ שמתקבל עשוי להיות לא מאוזן. [BKOS2000] מראה שיטה לבניית צמתים נוספים כך שהעץ יהיה מאוזן ויכיל $O(n)$ צמתים ועלות ביצוע של איזון העץ הוא $O((d+1)n)$.

הפתרון ש- [H2007] מביא לבעיית מציאת המצולעים שמכילים את הנקודה הוא הבא:
עבור ריבוע σ נגדיר רשימה conflict_list שמכילה את כל המצולעים שחותכים את σ. כמו כן נגדיר פרמטר של אורך ה- conflict_list המקסימלי לריבוע. במידה שאורך של conflict_list חורג מפרמטר זה אז הריבוע יחולק לארבעה רבעים ויבנה conflict_list עבור כל אחד מהריבועים הבנים. עבור כל אחד מהריבועים ייבנה צומת ב-Quadtree.

בהינתן נקודה p יש למצוא את הריבוע σ ב-quadtree שמכיל את נקודה זו. לאחר מכן יש לעבור על כל אחד מהמצולעים שנמצאים ב- conflict_list של σ ויש לדווח את המצולעים שמכילים את הנקודה p.



איור 51: עץ שנבנה על מנת לענות על השאילתות.

איור 51 מבוסס על [H2007].

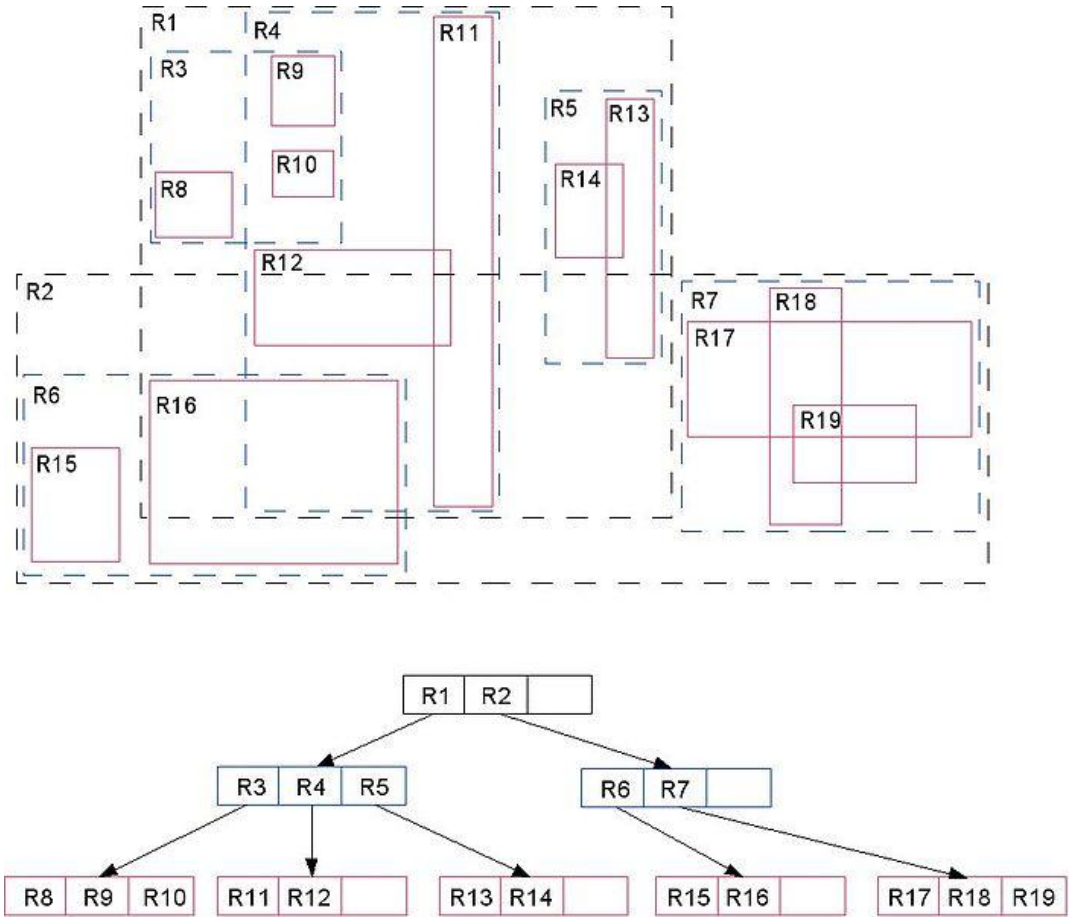
מציאת מלבן מסביב למלבן נתון.

עץ R (R-tree).

כמו במקרה של חיפוש רווחים מדובר בשאילתת הכלה.

אחד הפתרונות המקובלים הוא שימוש ב- R-tree [G1984], [BKSS1990], [AE1997].

בשיטה זו מבנה הנתונים עצמו מתבסס על עץ-B+ כאשר במפתח של כל צומת נמצא מצולע רב-ממדי. בצמתים הפנימיים המצולע המפתח של הצומת הוא בעצם המצולע הקטן ביותר MBR (Minimum Bounding Rectangle) המקיף את כל המצולעים שנמצאים בצמתים הבנים של אותו צומת. העלים, לעומת זאת, מכילים במפתח את המצולע (האינדקס) שמקיף את העצמים. מכיוון שמדובר בעץ מסוג B+-tree, העלים מצביעים על העצמים האמיתיים.



איור 52: תאור של R-tree.

איור 52 נלקח מ- http://zamotivator.dreamwidth.org/308533.html

באיור ניתן לראות שהמצולעים מכילים אחד את השני בצורה היררכית לפי מה שכתוב בעץ. למשל R1 מכיל את R3 ו-R4. R3 מכיל את R9 ו- R10, לעומת זאת R4 מכיל את R11 ו- R12.

עץ-R מוגדר בצורה הבאה [BKSS1990]:

בהינתן מספר M שהוא המספר המקסימלי של איברים שיכולים להיות בצומת, ו- $t \in [2, M/2]$,

אז לעץ-R התכונות הבאות:

1. לשורש לפחות שני בנים, אחרת הוא עלה.
2. לכל צומת פנימי בין t ל- M צמתים בנים, חוץ מהשורש שיכול להחזיק פחות מזה.
3. כל עלה חייב להחזיק בין t ל- M איברים.
4. כל העלים צריכים להופיע באותה רמה.

ישנם מספר שיקולי אופטימיזציה שעוזרים לייעל את החיפוש [BKSS1990]:

1. השטח של המלבן החוסם רצוי שיהיה קטן ככל האפשר, כך בזמן השאילתה תיחסך סריקה מיותרת של מסלולים נוספים בתוך העץ בעקבות חיתוך שווא בין המלבן החוסם של האינדקס השגוי ומלבן השאילתה.
2. החפיפה בין מלבני האינדקס צריכה להיות קטנה ככל האפשר, כך גם ייחסכו סריקות מיותרות.

3. היקף מלבן האינדקס צריך להיות קטן כפי האפשר. ידוע כי, עבור שטח נתון, המצולע בעל ההיקף הקטן ביותר הוא ריבוע. קל יותר להכניס ריבועים לתוך מלבן האב מאשר למלבנים האחרים.
4. גודל מבנה הנתונים צריך להיות קטן ככל האפשר.

תהליך החיפוש מוגדר בצורה הבאה [G1984]:

בהינתן עץ עם צומת השורש T ומלבן שאילתה S, אז:

1. יש לעבור על כל האיברים בצומת T ולמצוא את המצולעים שנחתכים עם S.
2. עבור כל מפתח שנחתך עם S יש לבצע חיפוש רקורסיבי לתת-עץ המתאים.
3. אם הצומת הוא העלה, אז עבור כל מפתח שנחתך עם S יש לדווח את העצמים שנשמרו ע"י מפתח זה ומקיפים את המלבן S.

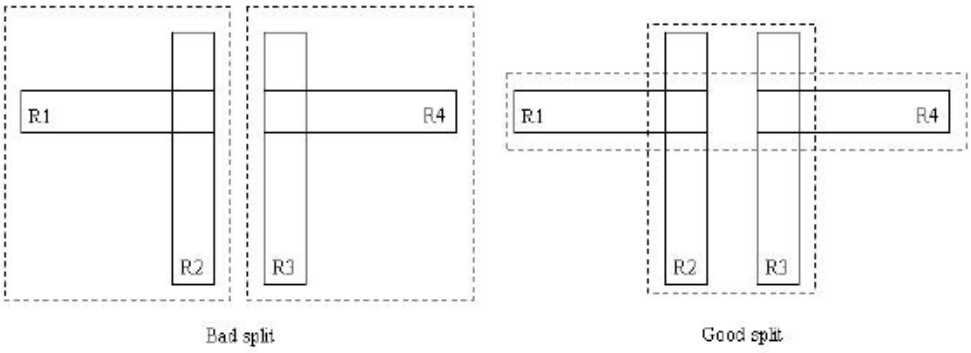
השאילתה שהיא מציאת מלבן מסביב לנקודה, היא מקרה פרטי של מציאת מלבן שמכיל מלבן S.

תהליך הוספת איבר חדש לעץ:

1. יש למצוא מלבן חוסם לעצם שנדרש להכניס אותו.
2. יש למצוא את העלה שאליו נדרש להכניס את המלבן. הרעיון הוא למצוא את העלה כך שאם מלבן המכיל את כל המלבנים שלו יכלול גם את המלבן החדש אז הוא יגדל בצורה מינימלית, כלומר יחס השטח לפני ואחרי ההכנסה יהיה מינימלי יחסית לשאר העלים. ברור שאם המלבן החדש מוכל במלבן קיים של עלה כלשהו, אזי ההוספה לא תגרום להגדלת השטח ותהיה אופטימלית.
3. יש להוסיף מצולע חדש לעלה, ואם גודלו של העלה חורג מ-M, אז צריך לעשות פעולת פיצול העולה בדומה לעץ-B, יתכן שכתוצאה מהפיצול כל העץ יגדל ברמה אחת.
4. יש לעדכן את המלבנים החוסמים של אבות הצומת (או צמתים במקרה של הפיצול) שהשתנו.

כאשר המצולע החוסם משתנה בעקבות הוספת בן חדש, האלגוריתם מבצע מחיקה של האינדקס הקיים, חישוב מצולע חדש והכנסתו מחדש ברמה זו.

הפעולה הכבדה ביותר שמתבצעת במבנה הנתונים היא פיצול הצמתים split. כעיקרון, פעולה זו היא חלק מהפעולות של עץ-B. כאשר אחד הצמתים מגיע למכסת הקיבולת שלו, כלומר יש בו M איברים ונדרש להוסיף לו עוד איבר, אז נדרש לפצל את הצומת כך שכל ההגדרות של העץ יישמרו. במקרה של עצי-R יש גם להתייחס לשיקולי האופטימיזציה שהוזכרו קודם כגון שמירה על השטח המינימלי של כל אחד מה-MBR. לכן, אחת המטרות של פעולת הפיצול היא לקבץ את כל המלבנים לשתי קבוצות כך ששטחם של מלבן חוסם של כל אחת מהקבוצות יהיה מינימלי.



איור 53: דוגמא לפיצול.

איור 53 נלקח מ- [G1984].

הפתרון הטריוויאלי הוא לעבור על כל האפשרויות ולבחור את הרכב הקבוצות האופטימלי. פתרון זה יהיה בעל סיבוכיות מעריכית.

[G1984] מציע פתרון היריסטי PickSeeds שאמור לעבוד בסיבוכיות של $O(n^2)$, אך הוא אינו מבטיח אופטימליות בבחירת הקבוצות. הרעיון של האלגוריתם הוא למצוא את זוג המלבנים שיבזבזו הכי הרבה שטח אם יהיו באותה קבוצה ואז להפריד אותם לשתי קבוצות. בהמשך כל אחד מהמלבנים הנותרים ייבדק מול כל אחת מהקבוצות ויוכנס לקבוצה שהגדלת השטח תהיה מינימלית ואילו היה נכנס לקבוצה השנייה אז הגדלת השטח הייתה מקסימלית. בחירת שני המלבנים הראשונים נעשית לפי הנוסחא

$$d = area(R) - area(E1\,\mathrm{Re}ct) - area(E2\,\mathrm{Re}ct) \qquad (81)$$

כאשר E1 ו- E2 הם זוג מלבנים מתוך הקבוצה שיש לפצל ו- R הוא מלבן חוסם את זוג המלבנים האלו; יש לבחור את E1 ו- E2 כך ש- d יהיה מקסימלי.

[G1984] מציע גם אלגוריתם לינארי כאשר תהליך בחירת המלבנים הראשוניים של כל אחת מהקבוצות הוא עוד יותר חלש: בוחרים שני מצולעים המרוחקים ביותר בכל אחד מהממדים.

[BKSS1990] מזכיר אלגוריתם פיצול לינארי נוסף שנקרא Greene Split. אלגוריתם זה מתבסס על בחירה של ציר החלוקה בין הקבוצות.

הביצועים של מבנה נתונים זה הם [ABHY2004]:

פעולה	עלות
גודל מבנה הנתונים	$\Theta(N/M)$
זמן ביצוע שאילתה	$O(\log_M N + k)$
זמן ביצוע הכנסה/מחיקה	$O(\log_M N)$

כאשר N היא כמות המלבנים הנשמרים במבנה הנתונים, M הוא המספר המינמלי של מלבנים בצומת, ו- k היא כמות המצולעים שישלפו בשאילתה.

הפתרון של מציאת המצולעים, המכילים את מצולע השאילתה, מבצע את שאילתת החיפוש הבסיסית כפי שהוגדרה לעיל ותוארה ב- [G1984].

ניהול מידע עבור מטרות מוטסות – Nearest neighbor

Voronoi diagram

כפי שראינו בפרקים הקודמים, אחד המרכיבים של תהליך העקיבה הוא תהליך שיוך של מדידות למסלולים הקיימים. מידת היעילות של השיוך משפיעה בצורה מכרעת על הביצועים של תהליך העקיבה, במיוחד אם מדובר באלגוריתם MTT. ראינו שאחת הדרכים הנפוצות לבחירת המדידה היא שימוש בשיטת Nearest Neighbour (NN). ההגדרה של בעיית NN היא הבאה [AMNSW1994]:

בהינתן קבוצה S של n נקודות במרחב המטרי X, נדרש לבצע עיבוד מקדים כך שבהינתן נקודת שאילתה $q \in X$ ניתן לדווח בצורה יעילה על הנקודה השייכת ל- S הקרובה ביותר ל- q. בעיה זו מוכרת גם בשם של בעיית סניפי הדואר.

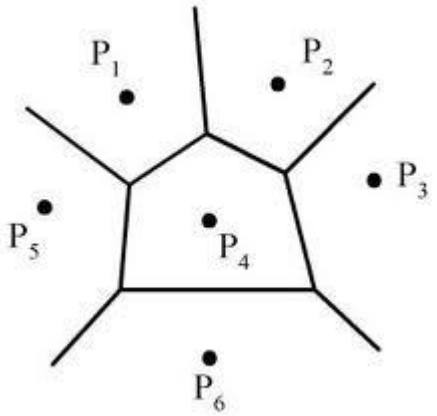
אחד הפתרונות לבעיה זו הוא שימוש בדיאגרמת וורונוי [L1982], [ABHY2004].

בהינתן קבוצת נקודות $S = \{q_1, q_2, ..., q_N\}$ במישור R^2 כאשר המרחק בין שתי נקודות מוגדר כ-

$d_p(q_i, q_j) = \sqrt{(q_i.x - q_j.x)^2 + (q_i.y - q_j.y)^2}$ או

$d_\infty(q_i, q_j) = \max(|q_i.x - q_j.x|, |q_i.y - q_j.y|)$. נגדיר את $h(q_i, q_j)$ כמקום הגאומטרי של כל הנקודות שקרובות יותר ל- q_i מאשר ל- q_j וזה יהיה חצי המישור שמוגדר ע"י

$B(q_i, q_j) = \{r \mid d(q_i, r) \le d(q_j, r)\}$. ואם כך המקום הגאומטרי של כל הנקודות שקרובות יותר ל- q_i מאשר לשאר הנקודות ב- S הוא $V(i) = \bigcap_{i \neq j} h(q_i, q_j)$ וגבולותיו יהיו הקווים שחוצים בין חצאי המישורים. הקדקודים של מצולע שמתקבל ייקראו נקודות וורונוי והקשתות ייקראו קשתות וורונוי.



איור 54: דוגמא לדיאגרמת וורונוי עבור 16 נקודות.

איור 55 נלקח מ-

http://par.cse.nsysu.edu.tw/~homework/algo01/8834809/VoronoiD.htm

האלגוריתם למציאת הנקודה הקרובה לנקודה p הוא:

- לזהות את המצולע pl שאליו שייכת הנקודה.
- להשוות את המרחקים מהנקודה p לכל הנקודות q שנמצאות במצולעים השכנים של pl.
- לבחור את הנקודה q שהמרחק ממנה ל- p הוא מינימלי.

[L1982] לא מתאר את הדרך למציאת המצולע שמכיל את הנקודה, אבל ניתן להשתמש בשיטות שתוארו למעלה, למשל שימוש בעץ-R.

לפי [L1982]:

פעולה	עלות
גודל מבנה הנתונים	$O(N)$
זמן ביצוע שאילתה	$O(\log N)$
זמן ביצוע הכנסה/מחיקה	$O(\log N)$
זמן בנייה של המבנה כולו	$O(N \log N)$

הדרך האלטרנטיבית שמוצעת ע"י [BKOS2000] היא שימוש ב- Quadtree. בשיטה זו בהינתן נקודת שאילתה q

- מזהים את העלה אליו הוא שייך.
- מנסים למצוא את ארבעת השכנים של התא (הצפוני, הדרומי, המזרחי והמערבי).
- בוחרים את השכן שהמרחק ממנו עד לנקודה q הוא הקטן ביותר.

[BKOS2000] טוען שאפשר לעשות שאילתה הרצה ב- $O(d+1)$ כאשר $d = O(\log(\frac{s}{c}) + \frac{3}{2})$.

כפי שראינו בפרקים הקודמים, גם המדידות וגם המסלולים מיוצגים גאומטרית ע"י האליפסה בגלל שגיאות המדידה. במקרה כזה נדרש למצוא את השכן הקרוב ביותר כאשר לפחות אחד הצדדים מיוצג ע"י אליפסה.

אם נניח שכדי שהמדידה תהיה רלוונטית למסלול, האליפסות של המסלול ושל המדידה צריכות להיחתך. ניתן להציע את השיטה הבאה:

- נבנה מלבן חוסם לאליפסות של המסלולים.
- נשמור את כל המלבנים בעץ-R.
- עבור כל אליפסת שאילתה נבנה מלבן חוסם.
- נשלוף את כל המלבנים שחותכים את מלבן השאילתה.
- נבחר את הזוג הקרוב ביותר המורכב מאליפסת השאילתה ואחת האליפסות שיישלפו.

ניתן להתחשב בהתקדמות הצפויה של המטרה כדי להגדיל את מלבן חוסם.

אפשרות אחרת היא להשתמש ב- kd-tree ולבצע שאילתת חלון מסביב למדידה ושוב לבחור את האליפסה הכי קרובה בין התוצאות המתקבלות.

7 סיכום

כפי שתואר בעבודה זו, בעיית העקיבה אחרי מטרות זזות הינה בעיה מורכבת ממספר שלבים:

- גילוי ואיכון.
- שיוך המדידה למטרות ידועות.
- עקיבה וניהול המטרות.

כל אחד מהשלבים האלו נדרש להתמודד עם המציאות שבה, מצד אחד, אמצעי המדידה הם מוגבלים ביכולת הדיוק שלהם, ולעתים נמצאים בתוך הקונפליקט בין חוסר גילוי של המטרות לעומת גילוי יתר והמצאת התראות שווא (False Alarm – Miss Detect). מצד שני, ישנם משאבים מוגבלים שמכתיבים ספים עליונים ליכולות חישוב, דיוקי מדידה וכו'. ניתן, אומנם, בהשקעת משאבים משמעותיים, לשפר את הספים, אבל אז המערכת תהפוך ללא כלכלית ובסופו של דבר תישאר בגדר פתרון תאורטי ובכל מקרה לא ניתן להגיע למצב שבו לא תהיינה שגיאות בכלל, בגלל שכל מודל של תהליך הינו קירוב למציאות ואינו מסוגל לתאר בצורה מלאה את התהליך. לדוגמא, במקרה של תהליך איכון המודל המתואר, מדובר באיכון דו-ממדי. במציאות מדובר במדידות לא מדויקות: הטעויות במדידה נובעות מאי דיוקים בפרמטרים רבים כגון, מיקום החיישנים, מהירות האור שמשתנה בתווך שונה, הפרעות, שגיאות באמצעי מדידה ורבים אחרים.

המטרה של המודל היא לתאר את התהליך בצורה קרובה ביותר למציאות, מחד, אבל מאידך, להישאר פשוטה מספיק כדי לממש אותה באמצעים העומדים לרשות המפתחים. התוצאה היא שהמודל הוא סטטיסטי ולכן יש לו מגבלות דיוק. מכיוון שהמערכת בנויה כתהליך בעל מספר שלבים, כל שגיאה שמתרחשת בשלב מוקדם, עוברת לשלב הבא ומשפיעה לרעה על התוצאות. בגלל כל הסיבות שלעיל, תוצאת תהליך העקיבה היא נתון הסתברותי ויש סיכוי שתדווח תוצאה שגויה. תוצאת האיכון היא: שערוך המקום וגם אליפסת האי-הודאות, כלומר אזור במרחב שהסתברות שהמטרה תהיה בתוכו גבוהה מסף מסוים. אם נגדיל את האזור עד אין סוף נוכל לומר בודאות שהמטרה נמצאת בתוך האזור, יחד עם זאת יעילותה של קביעה זאת די נמוכה. אם נגדיר מגבלה על גודל האזור, בעצם נגדיר הסתברות של שגיאת האיכון, כלומר נצטרך לקבל את העובדה שבאחוז מסוים מהתוצאות המטרה לא תהיה מוכלת באליפסה המדווחת על כל המשמעויות של שגיאה שכזו. בדרך כלל המודל של שגיאת המדידה הוא רעש גאוסיאני לבן, כלומר מתפלג בצורה נורמלית ללא סטייה (unbiased). שימוש במודל זה נובע משתי סיבות:

1. לפי משפט הגבול המרכזי, סכום משתנים מקריים בעלי התפלגות כלשהי (כל משתנה יכול להיות בעל התפלגות אחרת) שואף להתפלג בצורה נורמלית. לכן ניתן להניח שתהליך מקרי שמורכב מהרבה משתנים מקריים בעלי התפלגויות לא ידועות בסופו של דבר יתפלג בקירוב בצורה נורמלית.
2. שימוש בהתפלגות נורמלית נוח ביותר בחישובים בגלל שהפעלת פעולה לינארית על משתנה נורמלי משאירה את התוצאה להתפלג בצורה נורמלית.

פתרון בעיית האיכון מבוסס על שיטת Maximum Likelihood, האומרת שהמטרה צריכה להימצא בנקודה בעלת ההסתברות הגבוהה ביותר, לכן על מנת לאכן את המטרה יש לפתור

מערכת משוואות לא לינאריות, שמתארות את הסתברות הקליטה בזוית שדווחה ע״י כל חישן. לכן, הנקודה שהיא מרכז האליפסה, היא בעצם הנקודה עם ההסתברות הגדולה ביותר שהמטרה תמצא בה.

מערכת עקיבה מפעילה סריקה בעזרת החישנים ובסוף כל שלב של סריקה מתקבלות מספר מטרות. בשלב זה עליה לבצע שיוך (association) של המדידה למטרות הקיימות. בעיה זו אינה פשוטה, בגלל הן ריבוי המטרות והן הימצאות של מפריעים (clutter). איכות שיוך המדידות למטרות הינה קריטית בשל האופי האינקרמנטלי של תהליך העקיבה: טעות במחזור i של השיוך תגדיל את ההסתברות לטעות במחזור $i+1$. בעבודה זו מוצגים מספר פתרונות, החל מפתרון מציאת השכן הקרוב ביותר וכלה בשיטת MHT שבודקת מספר השערות ומחליטה על השיוך רק בשלב מאוחר יחסית אחרי מספר מחזורי סריקה. ההנחה היא שאם השערה מסוימת ביצעה שיוך בצורה מוטעית, אז תוך מספר מחזורים מועט ניתן יהיה לגלות את זה וההשערה תיפסל. כפי שכמות המחזורים שנשמרים לאחור תגדל, כך גם תשתפר האיכות של השיוכים. מצד שני, כמות ההשערות גדלה בצורה מעריכית, לכן נדרש להגיע לפשרה מסוימת בין איכות השיוך ליכולת החישובית של המערכת.

השלב הבא במערכת העקיבה הוא בעצם השלב של העקיבה בפועל. בגלל הסיבות שתוארו קודם, ברור שמדידה ששויכה למטרה לא מתארת בצורה מדויקת היכן המטרה נמצאת. מטרות העקיבה הן שתיים:

1. ״להחליק״ את תוצאות המדידות למסלול שמתאר בצורה האופטימלית את תנועת המטרה.
2. להיות מסוגלים לשערך היכן המטרה צפויה להימצא במחזור הבא של הסריקה.

יכולות אלו נותנות אפשרות למערכות כגון מערכת הנחיית הטיל להתכוונן בצורה המיטבית לכיוון המטרה וכך לשפר את הסתברות הפגיעה. השיטה שהוצגה בעבודה היא השיטה המקובלת של Kalman-Filter שעושה שערוך של המסלול כך שסכום ריבועי השגיאות (כלומר ההפרש בין הנקודה שנמדדה לנקודה המתאימה ששוערכה) יהיה מינימלי.

הרעיון של השיטה הוא להגדיר את המודל של התנועה הצפויה של המטרה, למשל מקובל להניח שהמטרה בדרך כלל זזה בקו ישר במהירות קבועה. כמו כן, יש להגדיר את השגיאה של המודל: ברור שמטרה יכולה לסטות מהקו שמוגדר ע״י המודל אבל לא בצורה חדה מדי. הטעות של המודל מגדירה מה היא מידת הסטייה המותרת. גם כאן יש קונפליקט תמידי: אם השגיאה המותרת היא קטנה, אז העקיבה לאחר מטרה מתמרנת כגון מטוס קרב תהיה לא נכונה. מצד שני, הגדלה של הטעות המותרת למודל תגרום לאי-ודאות בשערוך המסלול.

שיטה זו מקובלת בתחומים רבים כגון רפואה, כלכלה וכו׳. לשיטה זו יש מספר מגבלות:

1. כפי שנאמר קודם, היא תלויה מאוד באיכות השיוך של המדידות: שגיאה בשיוך המדידה עלולה לקלקל משמעותית את התוצאות.
2. יש קושי בהגדרת מודל מדויק שמתאר את התנועה. יתר על כן, יתכן שאותו עוקב יצטרך לעקוב אחר מטרות בעלות אופי תנועה שונה: מטוס נוסעים גדול שטס במהירות שיוט הקרובה למהירות הקול, מטוס קרב שמסוגל לטוס במהירויות הגבוהות בהרבה, וכמו כן

מטוס המבצע תמרונים מורכבים ומטוס קל שטס במהירויות של 200-300 קמ״ש. ברור שמודלים של תנועה של כל אחד מהמטוסים שונים מאוד.

3. הלינאריות של מודל התנועה. במידה שהמודל אינו לינארי, השיטה לא עובדת. ישנם מספר פתרונות שמוצגים בעבודה, כגון EKF שעושה לינאריזציה של המודל ועובד יפה בחלק מהמקרים, אבל במקרים אחרים מתבדר. שיטה נוספת היא UKF שבה שגיאת המצב של המטרה מתוארת לא בצורה אנליטית, אלא ע״י נקודות דגימה sample points.

החלק השני של העבודה מנסה להתמודד עם בעיית החיפוש של נקודה במרחב. ברור ששימוש במנגנון נוח של מציאת נקודות רלוונטיות מצד אחד, ופסילה מוקדמת של נקודות אחרות, מצד שני, תשפר משמעותית את תהליך השיוך וכך תשפר את כל מנגנון העקיבה.

בשלב הראשון מגדירים את בעיית החיפוש הכללי באופן פורמלי. הגדרה זו נעשית בעזרת תורת הקבוצות. כמו כן מוגדרים מדדי האיכות לצורך בדיקת השוואתית בין שיטות השונות . מדדים אלו תלויים בהגדרת מודל החישוביות, כלומר איך מיוצג זיכרון המחשב, כמה פעולות יש לבצע על מנת להגיע לתא מסוים בזיכרון וכו׳. מכיוון שבעיית החיפוש והמדדים מוגדרים בצורה פורמלית ניתן לשערך את החסמים של הביצועים האופטימליים. המשמעות של חסמים אלו ברורה כאשר מתכננים פתרון למערכות, שזמן החישוביות הוא קריטי.

בשלב השני מגדירים דרישות כלליות ממבני נתונים מרחביים וכמו כן מתארים בצורה פורמלית מספר שאילתות אופייניות. בהמשך מתוארים פתרונות לשאילתות כאשר נעשות הבחנות הבאות:

1. שאילתות במרחב חד-ממדי ביחס למרחב דו-ממדי.
2. שאילתות שמטרתן למצוא נקודה ביחס לשאילתות שמחפשות אזור.
3. מקום של אחסון הנתונים: זיכרון ראשי יחסית לזיכרון משני.

עבור כל אחד מהשאילתות מתואר פתרון וכמו כן מוגדרים מדדי האיכות שלו.

מטרת העבודה הזו היא לעשות סקירה של תהליך העקיבה מ״קצה לקצה״. בדרך כלל המאמרים מתמקדים בפתרון של אחת הבעיות שנמצאות בתהליך. האתגר של העבודה הוא לחבר בין הדיסציפלינות השונות: בעיות סטטיסטיות של איכון ועקיבה מצד אחד, ובעיות חיפוש וגאומטריה חישובית מצד שני, וכך להציג פתרון משולב. בכל פרק ניתן למצוא את הנוסחאות או את האלגוריתמים שניתן להשתמש בהם לצורך ביצוע העבודה. מצד שני העבודה מנסה לתאר את הדרך לקבלת נוסחאות אלו בצורה לא מורכבת, ככל האפשר על מנת לתת לקורא את האפשרות להבין מה מסתתר מאחורי הנוסחאות והאלגוריתמים. בעבודה זו נעשה ניסיון להביא למחנה המשותף את המאמרים שעליהם היא מתבססת, הן במסגרת חיבורם לרעיון אחד משותף והן בעזרת בניית שפה אחת של הגדרות המושגים והמשתנים שמופיעים בנוסחאות.

8 רשימת מקורות

[A1958] C. J. Anker, "Airborne direction finding – the theory of navigation errors", *IRE Trans.. Aeronautical and Navigational Electronics.*, Vol. ANE-5, No 4, 1958, pp. 199-210

[AAE2003] P.K. Agrawal, L. Arge and J. Erickson, "Indexing moving points", *Journal of Computer and System Sciences 66*, 2003, pp. 207-243

[ABHY2004] L. Arge, M. de Berg, H. Haverkort and K. Yi, "The priority R-tree: a practically efficient and worst-case optimal R-tree", *Proc. 2004 ACM SIGMOD Symposium on Principles of Database Systems*, 2004, pp. 47–57

[AE1997] P.K. Agrawal and J. Erickson, "Geometric range searching and its relatives." *Advances in Discrete and Computational Geometry*, pages 1-56. American Mathematical Society, 1998

[AHU1974] A.V. Aho, J.E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms", Addison-Wesley, Reading, MA, 1974

[AM1993] S. Arya and D. Mount, "Approximate nearest neighbor queries in fixed dimensions", *Proc 4th Annual ACM-CIAS Symposium on Discrete Algorithms*, 1993, pp. 271-280

[AMNSW1994] S. Arya, D. Mount, N. Netanyahu, R. Silverman and A. Wu, "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions", *Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms",* 1994, pp. 573-582

[B1995] Y. Bar-Shalom, "Multitarget-multisensor Tracking: Principles and techniques", 1st edition, Y. Bar-Shalom, 1995

[B2003] S.S. Blackman, "Multi-hypothesis tracking for multiple target tracking", *IEEE A&E Systems Magazine*, Vol. 19, No 1, Jan 2003, pp. 5 -18

[BD2001] P. Bickel and K.A. Doksum, "Mathematical Statistics: Basic Ideas and Selected Topics", Vol. 1, 2nd edition, Prentice-Hall, 2001

[BKOS2000] M. de Berg, M. van Kreveld, M. Overmas and O. Schwarzkopf, *"Computational Geometry: Algorithms and Applications"*, 2nd edition, Springer, 2000

[BKSS1990] N. Beckmann, H.P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles", *Proc ACM SIGMOD International Conference on Management of Data",* 1990, pp. 322-331

[BM2007] S.A. Banani and M.A. Masnadi-Shirazi, "A new version of unscented Kalman filter", *Proc. World Academy of Science, Engineering and Technology 26*, 2007, pp. 192- 197

[BS2008] P. Bezousek and V. Schejbal, "Bistatic and multistatic radar systems", Technical Report, Univ. of Pardubice, 2008

[C1984] K.L. Clarkson, "Algorithms for closest-point problems", Ph. D. thesis Stanford University, CA, 1984

[C1990] B. Chazelle, "Lower bounds for orthogonal range searching",*Journal of Association for Computer Machinery*, Vol. 37, No3, July 1990, pp. 439-463

[C1995] B. Chazelle, "Lower bounds for off-line range searching",*Proc. 27 Annual ACM Symposium on Theory of Computation*, 1995, pp. 733- 740

[CLR1990] T.H. Cormen, C.E. Leiserson and R.L. Rivest, "Introduction to Algorithms", MIT Press and McGraw-Hill,1990

[EM2001] P.J. Escamilla-Ambrosio and N. A. Mon, "Hybrid Kalman filter-Fuzzy logic architecture for multisensor data fusion", *Proc. of the 2001 IEEE International Symposium on Intelligent Control Mexico City*, Mexico 2001,364 – 369.

[G1984] A. Guttman, "R-Trees: a dynamic index structure for spatial searching", *Proc. 1984 ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47–57

[G1995] M. Gavish, "Source localization and data fusion from passive arrays", Ph. D. Thesis Tel-Aviv University, Tel-Aviv, 1995

[G2009] R. M. Gray, "Entropy and Information Theory", Springer Verlag, 2009

[GG1997] V. Gaede, and O. Günther, "Survey on multidimensional access methods," Technical Report, Department of Economics and Business Administration, Humboldt University Berlin, revised version, 1997

[H2007] S. Har-Peled, "Geometric Approximation Algorithms", http://valis.cs.uiuc.edu/~sariel/teach/notes/aprx/book.pdf, 2007.

[HL2001] D.L. Hall and J. Llinas, "Handbook of Multisensor Data Fusion", CRC Press, 2001

[HCCL2003] C. Hu, W. Chen Y. Chen and D. Liu,"Adaptive Kalman Filtering for Vehicle Navigation",*Journal of Global Positioning Systems",*Vol. 2, No. 1, 2003, pp. 42-47

[J2009] J. M. Jose, "Source localization and data fusion from passive arrays", Master Thesis Lulea University of Technology, Prague, 2009

[K1960] R.E. Kalman, "A new approach to linear filtering and prediction problems", *Transactions of the ASME – Journal of Basic Engineering*, No. 82 (Series D). (1960), pp. 35-45.

[K1993] S. Kay, "Fundamentals of statistical signal processing: estimation theory", 1st edition, Prentice-Hall, 1993

[K2007] D. Koks, "Numerical calculation for passive geolocation scenarios", Tech. Report, Defence science and technology organization, Edinburg, Australia,2007

[KNS2004] P. Konstantinova, M. Nikolov and T. Semerdjiev, "A study of clustering applied to multiple target tracking algorithm", *Proc. CompSysTech*, 2004, pp. 22-1 – 22-6

[L1982] D.T. Lee, "On k-nearest neighbor Voronoi diagrams in the plane", *IEEE Transactions computer,* Vol. 31, No 6, June 1982, pp. 478-487

[LLHH2005] H. Li, H. Lu, B. Huang and Z. Huang, "Two ellipse based pruning methods for group nearest neighbor queries", *Proc. GIS'05*, 2005 pp.192-199

[M1979] P. S. Maybeck, "Stochastic Models, Estimation and Control", Vol 1, Academic Press, Inc.

[MBCC1999] S. Mori, W. Barker, C. Y. Chong and K.C. Chang, "Track association and track fusion with non-deterministic target dynamics", *Proc. Fusion99 Sunnyvale, CA",* 1999, pp. 1-8

[MR2009] R. Martinelli and N. Rhoads, "Predicting market data with a Kalman filter", Tech Report, Haiku Laboratories, 2009

[NHS1984] J. Nievergelt, H. Hinterberger and K.C. Sevcik, " The Grid File: An Adaptable, Symmetric Multi-Key File Structure", *ACM Transactions on Database Systems*, vol. 9, no. 1, Mar. 1984, pp. 38-71

[OTKTF2009] V. Oikonomou, A. Tzallas, S. Konitsiotis, D. Tsaliakis and D. Fotiadis, "The use of Kalman filter in biomedical processing", *Kalman Filter Recent Advances and Applications*, 1st ed., M Moreno and A Pigazo, ed., I-Tech, Vienna, Austria, 2009, pp. 164-180

[P1990] W. Pugh, "Skip lists: a probabilistic alternative to balanced trees", *Communication of ACM*, vol. 33, no. 5, Jun. 1990, pp. 668 – 676

[P2006] G.K Parischa, "Kalman filter and its economic applications*"*, Technical Report, Univ. of California, 2006

[R1979] D.B. Reid, "An algorithm for tracking multiple tragets", *IEEE Transactions on automatic control,* Vol. 24, No 6, Dec 1979, pp. 843 -854

[R2004] M.I. Ribeiro, "Gaussian probability density functions: properties and error characterizations", Tech Report, Institute for Systems and Robotics, Lisboa 2004

[R2004] M.I. Ribeiro, "Kalman and extended Kalman filters: concept, derivation and properties", Tech Report, Institute for Systems and Robotics, Lisboa 2004

[RG2003] R. Ramakrishnan and J. Gehrke, "Database Management Systems", 3d edition, McGraw-Hill, 2003

[RLT2005] K. Roy, B. Levy and C.J. Tomlin, "Target tracking and estimated time of arrival (ETA) prediction for arrival aircraft", *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, pp. 1-22

[S1947] R. J. Stansfield, "Statistical theory of DF fixing", *IEEE Proc Radar, Sonar & Nav.*, vol.94, no. 15, Mar. 1947, pp. 762-770

[S1998] G.W. Stimson, "Introduction to Airborne Radar", 2nd edition, SciTech Publishing Inc., 1998

[SBC1999] L.D. Stone, C.A. Barlow and T.L Corwin, "Bayesian Multi-target Tracking", Artech House, Inc., Norwood,1999

[SC2000] C.B. Shim and J.W Chang, "Spatio-temporal representation and retrieval using moving object's trajectories" , *Proc. 2000 ACM Workshops on Multimedia*, New York, NY, 2000, pp. 209-212

[SS1971] R.A. Singer and J.J Stein, "An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems" , *Proc. 1971 IEEE Conference on Decision and Control,* Miami Beach, FL, 1971, pp. 171-175

[ST1986] N. Sarnak and R. Tarjan, "Planar point location using persistent search trees" , *Communication of ACM*, vol. 29, no. 7, Jul. 1986, pp. 669 – 679

[T1984] D.J. Torrieri, "Statistical theory of passive location systems", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-20, no 2, Mar. 1986, pp. 183-198

[W1971] L.H. Wegner, "On the accuracy analysis of airborne techniques for passively locating electromagnetic emitters", Technical Report, Rand Corp. available from National Technical information Service, 1971

[WB2006] G. Welch and G. Bishop, "An introduction to the Kalman filter", Technical Report, Dept. of Comp. Science, Univ. of N. Carolina at Chapel Hill, 2006

9 Abstract

The ultimate objective of a tracker of airborne targets is to estimate the movement trajectory of the objects that dwell in the area of the interest of the air space. The results of the tracker will support the decisions that must be taken by the systems or the operators that are in charge of the air traffic in the area of interest. The tracker system has several challenges to deal with such as: the capability to detect the object in the clutter area, location of the objects and the actual estimation of the trajectory of each object. On the other hand, the system should be efficient enough in order to comply with various processing time limitations and still to be cheap enough in order to be competitive.

Usually, the tracker system contains subprocesses such as sensors, locator and tracker. Since each of them is error prone due to its physical limitations, it is impossible to find exact solutions and it is necessary to estimate approximate results based on statistical methods.

One of the main parts of the process is the association of the located detections during the system scan to already known targets. The association is done by searching for the best match between detections to the candidate tracks. It is obvious that the optimization of the search will improve the total performance of the tracker system. The problem of "searching for an object" has been studied for many years and the lower bounds for the search algorithm have been calculated. The field of computational geometry presents classes of data structures, called spatial data structures, which provide an efficient way for object search in multidimensional spaces.

The objective of this work is to present the tracker system by reviewing several solutions for location and tracking algorithm, which were developed during the last decades. The second part of the paper surveys several spatial data structures that may be used as indexes for the searching problems. For each of the data structures its performance is presented.

Table of contents

The Open University of Israel

Department of Mathematics and Computer Science

The tracking of airborne targets by using spatial data structures

Thesis (Final Paper) submitted as partial fulfillment of the requirements towards an M.Sc. degree in Computer Science

The Open University of Israel

Computer Science Division

By

David Rozentur

306359985

Prepared under the supervision of Dr. Jack Weinstein

2011