

הטכניון - מכון טכנולוגי לישראל
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY

הפקולטה להנדסת חשמל
המעבדה לבקרה רובוטיקה ולמידה חישובית



Control Robotics and Machine Learning Laboratory

דוח פרויקט : שיתוף פעולה בין רחפנים

שיתוף פעולה בין רחפנים לביצוע משימת הרמה של חבילה והעברתה במרחב

מגישים

יניב חסידוף

יונה רובל

מנחה

איל טייטלר

סמסטר אביב

2022

תוכן עניינים

3	רשימת איורים, תמונות ותרשימים
3	מטרת הפרויקט
3	רקע
4	הגדרת המשימה
4	תיאור המערכת
5	תקשורת
6	פעולות המערכת
7	לוקליזציה
7	ראייה ממוחשבת - זיהוי עצמים
9	ביצוע מסלול
9	שמירה על המרחק היחסי בין הרחפנים
9	מציאת מרחק יחסי בין רחפנים
9	תכנון מסלול
10	ביצוע המסלול
12	תוצאות
13	סיכום ועבודה להמשך
13	רשימת מקורות

רשימת איורים, תמונות ותרשימים

מטרת הפרויקט

מטרת הפרויקט הינה שילוח של חבילה באמצעות שני רחפנים בצורה אוטונומית לפי הגדרות מפעיל. את המסלול יש לבצע בצורה אוטומטית תוך שמירה על החבילה וללא התנגשויות.

רקע

רחפן הוא כלי מעופף נשלט מרחוק. מבנה הרחפן כולל ארבעה או יותר מנועים חשמליים המחוברים למדחפים מכוונים כלפי מעלה. השליטה במהירות הסיבוב של על מדחף בנפרד נותנת לרחפן יכולת לנוע בצורה תלת מימדית בסביבה כמעט ללא הגבלה. לרוב לגוף הרחפן מחוברים חיישנים כגון מצלמות, GPS, ג'ירוסקופ ועוד. יכולת משמעותית הקיימת אצל מגוון רחפנים כיום היא שליטה אוטונומית - היכולת לבצע משימות ללא שליטה על ידי מפעיל אנושי.

השימוש ברחפנים צובר תאוצה בשנים האחרונות במגוון תחומים - בין אם בתחום הצילום והמשלוחים, או בתעשיות ובחקלאות. בכל התחומים הללו האפשרות להטיל על רחפן משימות שבעבר רק עובד אנושי יכול לעשות גורמת להפחתה משמעותית בדרישות כוח אדם ומשאבים. עם זאת, לרחפן מגבלות משלו - לרחפן לרוב יש סוללה קטנה וכתוצאה מכך זמן מועט לבצע את משימתו לפני שנדרש להיטען מחדש. בנוסף, לרחפנים יכולת נשיאה קטנה יחסית. חסרונות כמו אלו מונעים מרחפנים לבצע מגוון משימות רחב יותר שעדיין דורשות מפעיל אנושי תוך שימוש בכלים אחרים.

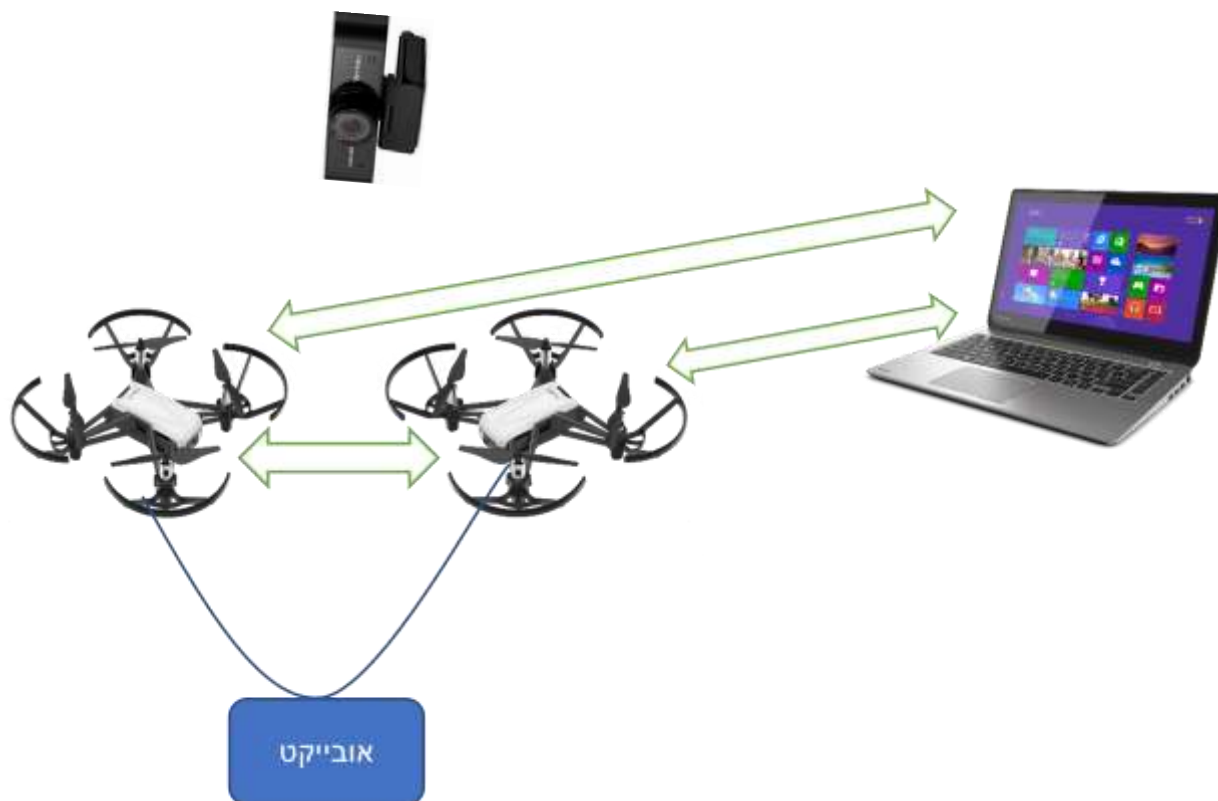
בפרויקט שלנו נרצה להראות כי על חלק ממגבלות אלו ניתן להתגבר בעזרת שיתוף פעולה בין רחפנים. מטרת הפרויקט היא להשתמש במספר רחפנים כדי לבצע משלוח אווירי בצורה אוטונומית. שימוש במספר רחפנים להרמה של חבילה אחת ביחד מפצה על כושר הנשיאה המוגבל של כל רחפן לחוד. בנוסף, שילוח בצורה זו יתרום לטווח טיסה ארוך יותר בהשוואה לרחפן יחיד שנדרש לסחוב חבילה זהה. לשיטה זו יכולים להיות שימושים מגוונים בין אם בתחום השילוח האוטונומי ועד ניהול מלאי במחסנים, שינוע חומרים בתוך אתר בניה ועוד רבים.

הגדרת המשימה

בתרחיש אותו נבחן הרחפנים מתחילים מחוברים למטען ונדרשים להסיע אותו ליעד מבוקש לפי פקודות הניתנות ממחשב מרכזי. על המחשב קיים ממשק דרכו ניתן להגדיר לרחפנים פקודות כגון המראה, נחיתה וטיסה לנקודה מסוימת.

המסלול אל היעד יתוכנן במחשב, ופקודות בהתאם יועברו לרחפנים כדי לבצע את המסלול. מנקודת מבטו של המשתמש הרחפנים יתפקדו כיחידה אחת הנעה במבנה ומבצעת מסלול. מבחינת אלגוריתם הבקרה כל רחפן בודד יצטרך לשמור על המיקום שלו במבנה ולתקן בהתאם, תוך כדי השלמת הפקודות. בכל רגע נתון תישמר לנו המשימה הנוכחית כפי שניתנה ע"י מפעיל, והרחפנים יפעלו לבצע את משימה זו. משימות ניתן להגדיר ולהרחיב בזמן אמת באמצעות ממשק גרפי.

תיאור המערכת

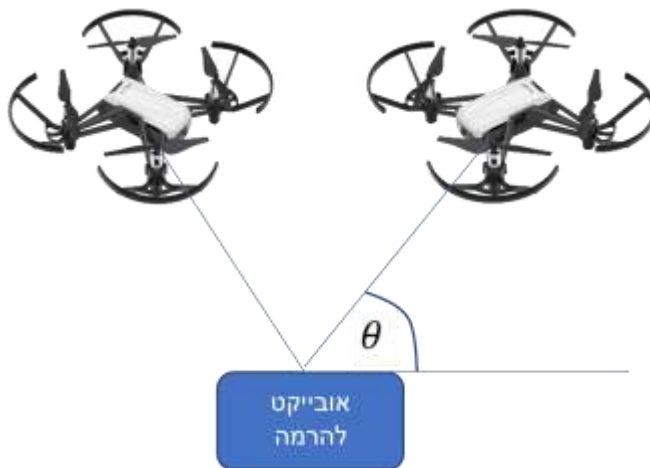


המערכת מורכבת מהחלקים הבאים:

מחשב שליטה מרכזי: תפקידו לבצע את החיבוריות לרחפני tellon לרשת התקשורת על בסיסה נעביר את המידע. הפקודות נשלחות בעזרת ספריית tello-python אשר מכילה פקודות בסיסיות (sdk) אשר ניתן לשלול לרחפן. בחיבור ראשון לרחפנים יש לקשר את התקשורת של ה- tello לרשת הרצויה. מבצעים זאת על ידי מציאת כתובות ה- IP המוקצות לכל רחפן וביצוע החיבור על ידי פקודת connect אשר מקבלת את כתובות ה- IP כארגומנטים. בהמשך נרחיב על מנגנון התקשורת.

שני רחפני tello: הרחפנים יקבלו מידע מן המחשב. כמו כן, נגדיר רחפן אחד להיות המוביל ורחפן שני להיות העוקב.

אובייקט להרמה: על האובייקט להיות במשקל מתאים ליכולות ההרמה של הרחפנים על מנת שהמשקל לא ישפיע על זווית הטיסה שלהם. משקל מקסימלי שכל רחפן יכול להרים הינו 70 גרם (נסמן משקל זה ב-m). נבצע חישוב כדי למצוא את המשקל המקסימלי שיוכלו להרים יחד:



מבצעים חישוב טריגונומטרי ומתקבל כי המשקל המקסימלי הינו:

$$2m \geq M_{\max}$$

ניתן יהיה להניח כי משקל החוט זניח.

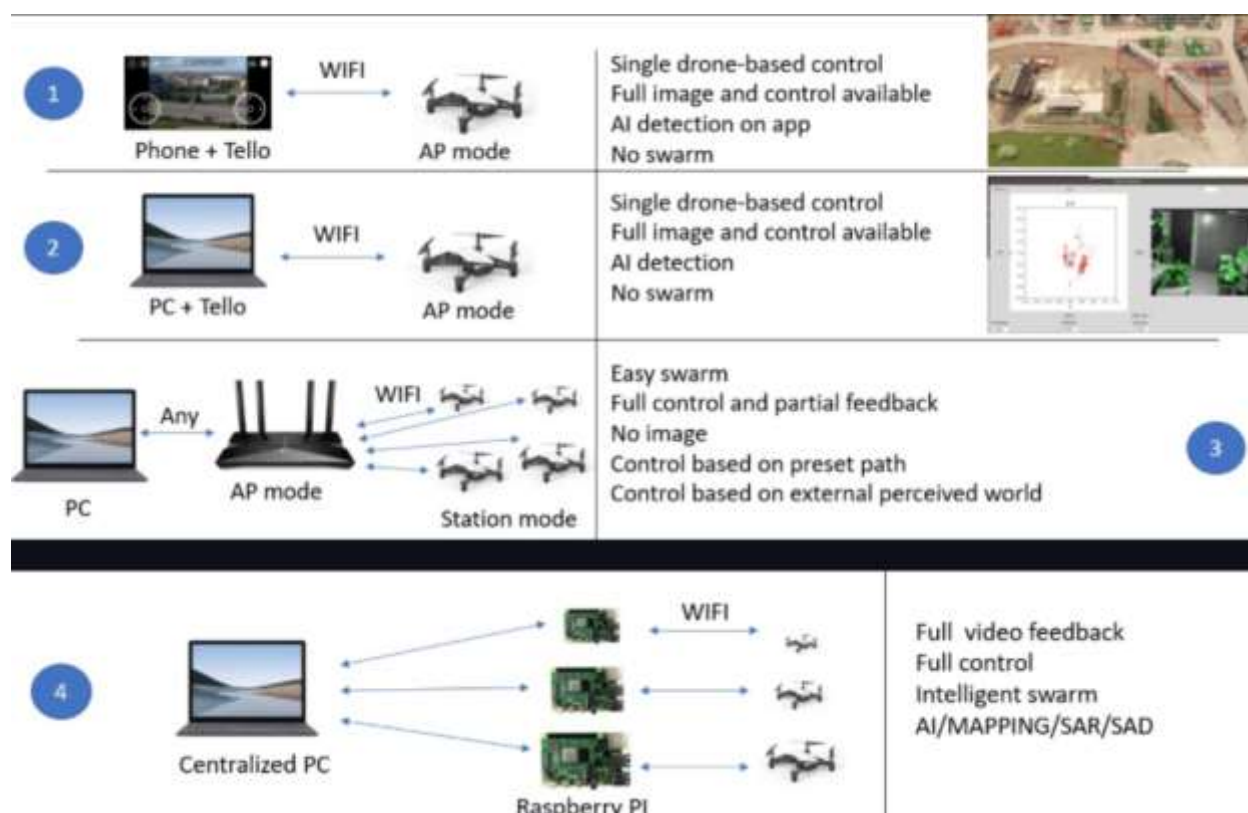
$$\frac{m \cdot 2}{\sin \theta} \geq M$$

מצלמת תקרה: מצלמה זו תהיה מודבקת לתקרה ותמוקם כך שניתן יהיה לראות שדה ראייה מספק. שדה הראייה יגדיר את גבולות הטיסה. באמצעות המידע מהמצלמה, נבחר את נקודת היעד אליה הרחפנים יצטרכו לשנע את החבילה, והגדרת מסלול הטיסה שלהם.

תקשורת

הרחפנים מקבלים ושולחים פקודות בעזרת wifi המעביר הודעות בפרוטוקול UDP. כדי להשתמש בתכונת swarm של הרחפנים, אותה תכונה המאפשרת שליחת וקבלת פקודות ממספר רחפנים במקביל, יש לבצע בחיבור ראשוני למערכת את הפעולות הבאות: יש לחבר את המחשב לרשת wifi של אחד

הרחפנים. לאחר מכן, בעזרת פקודה מה SDK של הרחפן, יש לבצע חיבור לנקודת גלישה אשר יכולה להתחבר למספר רשתות במקביל. לשם כך התחברנו לנקודת גלישה של הטלפון הנייד. יש לבצע את הפעולה עבור 2 הרחפנים כך שהתוצאה היא כך: 2 הרחפנים והמחשב מחוברים לאותה נקודת גלישה (במקרה שלנו נקודת גלישה של הטלפון הנייד). יש לציין כי במצב זה אין אפשרות לשדר וידאו מהרחפנים. מגבלה זו נובעת מפני שהוידאו מכל הרחפנים מקבל לאותו פורט (פורט 11111) ולמעשה מונע מהמחשב לקבל את התמונה. במהלך ביצוע הפרויקט ביצענו ניסוי אשר מנסה להתגבר על בעיית שליחת הוידאו. חיברנו למחשב מתאם רשת נוסף ובעזרת תכנות מקבילי חיברנו כל רחפן למתאם רשת אחר כך שהרחפן שולח ומקבל פקודות רק מהמתאם רשת שאליו הוא מחובר. באופן זה התקשורת מתבצעת בדומה לאופציה השנייה המופיעה בסכמת סוגי התקשורת המצורפת למטה, עבור כל רחפן באופן עצמאי. כלומר בדרך חיבור זו אין אפשרות להשתמש בתכונת ה swarm. שיטת עבודה זו בסופו של דבר הייתה איטית ורצף קבלת ושליחת הפקודות מהרחפנים גרם לחוסר יציבות וסנכרון המערכת. לכן העדפנו להשתמש בשיטה הקודמת ולבצע שימוש בתכונת swarm.



סכמת סוגי התקשורת האפשריים עבור רחפני טלו

פעולות המערכת

את ביצוע המשימה ניתן לחלק לארבעה מרכיבים:

- מציאת מיקום הרחפנים בזמן אמת
- מציאת המרחק היחסי בין הרחפנים בזמן אמת

- תכנון מסלול מהמיקום הנוכחי למטרה הרצויה
- ביצוע המסלול תוך עדכון מתמיד ומניעת התנגשויות

לוקליזציה



לוקליזציה בהקשר של פרויקט זה היא מציאת מיקום הרחפנים במערכת ייחוס מוגדרת. לרוב, רחפנים שמבצעים ניווט בשטח פתוח מתבססים על לווני GPS על מנת לאכן את מיקומם. מערכת הייחוס במקרה זה תהייה מערכת הצירים הגלובלית, המוגדרת בקואורדינטות של קווי אורך ורוחב.

במסגרת הפרויקט הוגבלנו לביצוע ניסויים אך ורק בתוך מבנים סגורים. בתוך מבנה אלו לווני אינו יכול לאכן את הרחפן. כמו כן הדיוק הנדרש בתוך חדר

הינו גבוה בהרבה משניתן להשיג להשיג ע"י GPS. על מנת להתגבר על בעיות אלו נדרשנו למצוא שיטה חלופית להגדרת מערכת ייחוס ומציאת הרחפנים בתוכה.

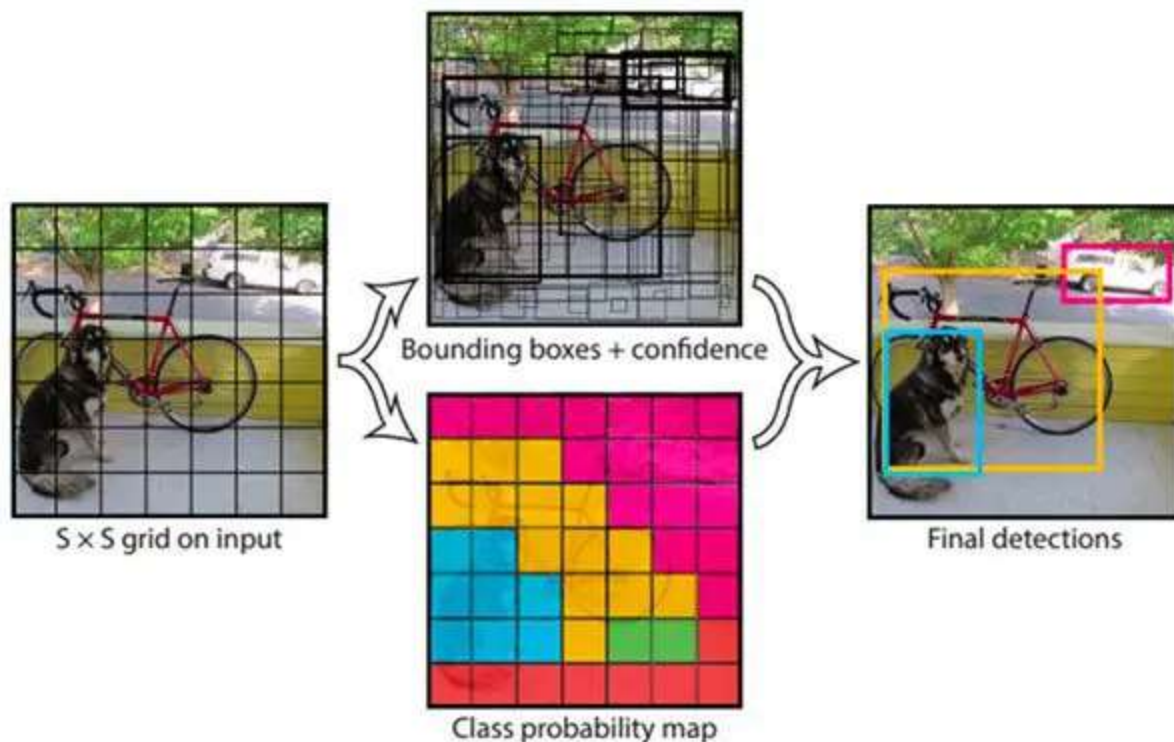
השיטה שבחרנו מבוססת על מצלמה חיצונית המוצבת בתקרת חדר הניסוי. את מערכת הצירים של המצלמה בחרנו בתור הפיקסלים המרכיבים את התמונה הנתונה.

ראייה ממוחשבת - זיהוי עצמים

את זיהוי מיקום הרחפנים בתמונה ניתן לבצע במספר רב של שיטות. השיטה שבחרנו לפרויקט היא באמצעות רשתות נוירונים עמוקות (Deep Neural Networks).

הרשת שבחרנו היא 5YOLOv.

עיקרון הפעולה של 5YOLOv הוא ביצוע מספר תת-הרצות של רשת הבנויה לסיווג תמונה, כאשר בכל פעם הקלט לרשת הפנימית הוא חלק מהתמונה המקורית. הגדרת החלקים המועברים לרשת הפנימית נעשה באמצעות Anchors (עוגנים), כאשר עוגן מוגדר להיות צורה מלבנית אשר משמשת כ'חלון נע' על פני התמונה המבצע דגימה של הפיקסלים הנמצאים בתוכו.



סכמת הפעולה של רשת YOLOv5

לשימוש המצלמה החיצונית הגדרנו את רוב העוגנים להיות מלבנים קטנים בעלי תדר דגימה גבוהה על מנת לזהות בעיקר אובייקטים קטנים, שכן הרחפנים נמצאים במרחק גדול יחסית ולרוב רוחבם בתמונה לא עולה על כמה עשרות פיקסלים.

עבור משקולות הרשת השתמשנו ברשת שאומנה מראש על זיהוי רחפנים, והמשכנו את האימון בעצמנו על כמה עשרות תמונות של רחפן ה Tello שברשותינו.



דוגמה להפעלת הרשת על רחפן טלו

במהלך הטיסה את מציאת מיקום הרחפנים ביצענו באמצעות זיהוי עצמים בתמונה. מכיוון שרשת YOLO מחזירה רק את המיקום של 2 רחפנים, עלינו לשייך כל מיקום לרחפן המתאים. לשם כך בכל צעד עדכון יתבצעו בשלבים הבאים:

- נניח כי קיימים לנו המיקומים הקודמים/התחלתיים של הרחפנים a,b
1. ניקח תמונה מהמצלמה.
 2. נכניס את התמונה כקלט לרשת זיהוי אובייקטים, ונקבל 2 מיקומים של רחפנים a',b'.
 3. ניצור וקטור בין 2 המיקומים a'b' ווקטור אנטי סימטרי b'a'. כל וקטור מייצג "שיבוץ" של אחד הרחפנים במיקומים שהתקבלו במצלמה. במקביל ניצור את באותו אופן וקטור בין המיקומים הישנים a,b.
 4. נחשב את הפרשי הזוויות בין 2 הוקטורים החדשים והוקטור הישן. את הוקטור הקרוב ביותר נבחר בתור הוקטור המייצג את מיקום הרחפנים האמיתי בתמונה.

ביצוע מסלול

מציאת מסלולים עבור מספר סוכנים תוך שמירה על אילוצי המרחק ביניהם היא משימה יחסית מורכבת ללא פתרונות זמינים לשימוש. את פתרון הבעיה מצאנו באמצעות רדוקציה לשתי בעיות נפרדות בעלות פתרון יחסית פשוט לכל אחת מהן - מציאת מסלול עבור סוכן יחיד, ושמירה על מרחק יחסי קבוע בין כל הסוכנים.

שמירה על המרחק היחסי בין הרחפנים

עבור פישוט תהליך תכנון המסלול הגדרנו כי הרחפנים יטוסו במבנה קבוע. את המבנה הגדרנו בתור צורה גיאומטרית פשוטה, כך שחרחפנים ימוקמו בחוקדוקדים - למשל שלושה רחפנים יטוסו במשולש, ארבעה בריבוע וכן הלאה. בעת תחילת המשימה נבחר את הרחפן הקדמי ביותר בכיוון הטיסה להיות המנהיג. במהלך הטיסה כל שאר הרחפנים יהיו מכוונים אליו כך שיוכלו לקלוט אותו במצלמה הקדמית. תפקידו של המנהיג הוא לבצע את המסלול. שאר הרחפנים גם הם מבצעים את המסלול אך בנוסף קיימת עבורם משימה נוספת - שמירה על מרחק וזווית קבועים ביחס למנהיג. במקרה של סטייה תוך כדי טיסה הרחפנים יבצעו תיקון ע"י שינוי מהירות בצירים הרלוונטיים.

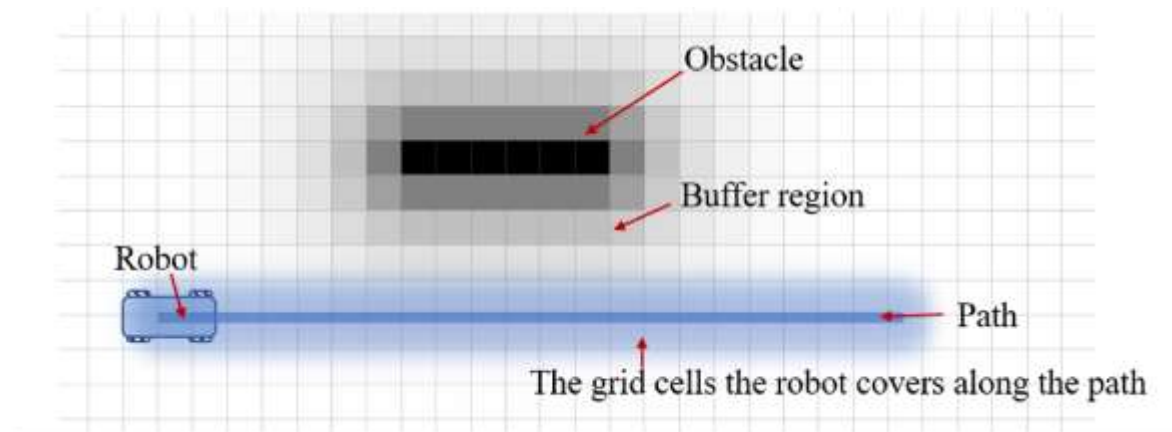
מציאת מרחק יחסי בין רחפנים

את הזווית היחסית בין הרחפנים ניתן לגלות באמצעות תשאול המצפן של כל רחפן בעזרת פקודות ה SDK. את המרחק היחסי בין הרחפנים מצאנו באמצעות המצלמה הקדמית של כל רחפן. את מהלך הטיסה תכננו כך שבכל עת כל הרחפנים יקבלו קו ראייה ישיר לרחפן המנהיג. במהלך ביצוע המסלול כל רחפן מבצע זיהוי תמונה למציאת הרחפן המנהיג, באופן דומה לשיטה בה השתמשנו לזיהוי הרחפנים במצלמה החיצונית. לפי גודל המנהיג ומיקומו בפריים חישבנו את מרחקו מהמצלמה ואת מיקומו היחסי.

תכנון מסלול

מכיוון שמעתיה ניתן להתייחס אל קבוצת הרחפנים בתור מבנה יחיד, את חיפוש המסלול ביצענו באמצעות אלגוריתמי חיפוש עבור סוכן יחיד. עבור הרחפנים שלנו בחרנו להשתמש באלגוריתם *RRT. אלגוריתם *RRT מבצע חיפוש רנדומלי על גרף באמצעות בניית עצים. אופי המסלול המתקבל הוא לרוב אוסף של קווים ישרים, שזהו אופי המתאים ביותר לרחפנים. מכיוון שהמסלול המתקבל מאלגוריתם זה מורכב מקווים ישרים וביניהם פניות, הוא יחסית יותר פשוט לביצוע תוך שמירה על מבנה. כמו כן הוספנו את האפשרות לבחור באופן ידני נק' הבונות מסלול, כך שהרחפנים יעקבו אחריו. אופציה זו מאפשרת למשתמש להתחשב באילוצים אחרים מלבד אורך המסלול (מספר פניות מועט, העדפה לאיזור מסוים, וכו').

בעיה נוספת שעלתה בעת חיפוש המסלולים היא ש RRT* אינו מתחשב בגודל מבנה הרחפנים. לשם כך נעזרנו באוגמנטציית הרחבה למפה. הרחבה בתמונה בינארית מבצעת עבור כל פיקסל בעל ערך '1' שינוי של הערכים בפיקסלים השכנים שלו מ'0' ל-'1'.



אוגמנטציית המפה לבניית חוצץ סביב מכשולים. התמונה לקוחה מ
<https://www.mdpi.com/1424-8220/21/3/961/html>

ביצוע המסלול

בתחילת הניסוי נציב את שתי הרחפנים בנקודות ההתחלה שלהם. נציב אותם על הרצפה בצורה כזאת שהרחפן המוגדר מוביל עומד לפני הרחפן העוקב, כך מצלמת הרחפן העוקב מכוונת אל המוביל. לאחר חיבור מוצלח לשני הרחפנים נשלח לשניהם פקודה להמראה. באמצעות אלגוריתם הזיהוי הרחפן העוקב מזהה את המוביל ומסמן סביבו מלבן. כאשר הרחפן המוביל מתחיל בטיסתו, המלבן סביבו ישתנה בהתאם (יגדל, יקטן או יסטה לצדדים, ביחס למרכז התמונה). תוך כדי טיסתו יתבצע חישוב על מנת לשמור את המלבן המסומן במרכז תמונתו של הרחפן העוקב. לשם כך ניתן פקודות טיסה לרחפן העוקב. לדוגמא כאשר המלבן יסטה ימינה ממרכז תמונתו, הרחפן יקבל פקודת טיסה כזו שתזיז אותו ימינה.

מכיוון שביצוע פניה באופן מתואם היא פעולה מורכבת כאשר מדובר במספר רחפנים עם חבילה, הגדרנו פרוטוקול מיוחד לביצוע פניות. בעת ביצוע פניה, על מנת לשמור על המיקום היחסי בין הרחפנים יבוצע האלגוריתם באופן הבא:

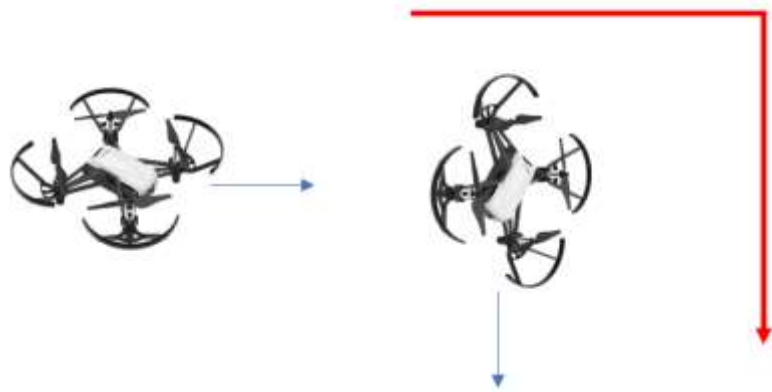
(1) הרחפן המוביל מבצע את פניה במקום בזווית הנדרשת, עם או נגד כיוון השעון:

כיוון הפניה



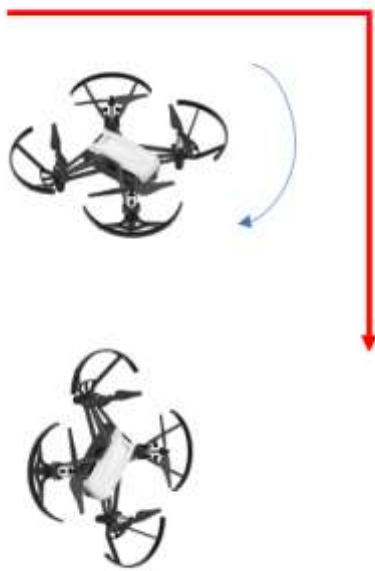
(2) שני הרחפנים ינועו קדימה במקביל למרחק של מטר:

כיוון הפניה

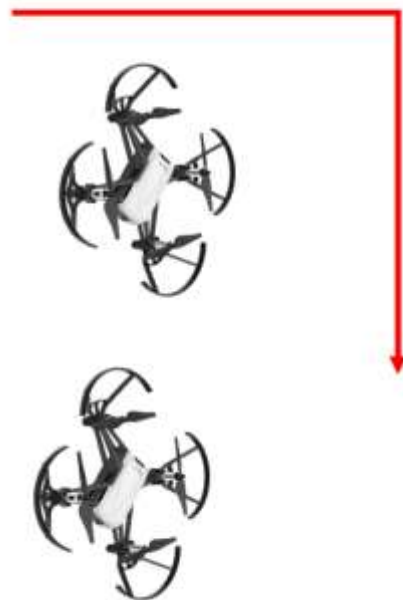


(3) הרחפן העוקב מבצע את הפניה הנדרשת כך שכעת המצלמה שלו מול החלק האחורי של הרחפן העוקב.

כיוון הפניה



כיוון הפניה



תוצאות



צילום מאחד הניסויים הכולל שילוח של חבילה באמצעות שני הרחפנים

ביצענו מספר ניסויים בחדר סגור בהם מתבצע מסלול שהוזן ידנית לרחפנים. בעת הניסוי הרחפנים הצליחו להמריא, לעבור בנקודות הרצויות ולבסוף לנחות תוך שמירה על מרחק יחסי, שלמות החבילה ובתיאום מלא. מכיוון שלא הצלחנו להפעיל את מצלמות הרחפנים בעת הניסוי התיאום מתבצע אך ורק על סמך הזיהוי במצלמה החיצונית.

בחלק מהניסויים נתקלנו בעת ביצוע פניות ובנחיתה בבעיות סנכרון. בעיות אלה נבעו מבעיות בפרוטוקול התקשורת עבור רחפני טלו, בהן אחד הרחפנים לא קיבל את הפקודה למרות השימוש הנכון בפעולות הסינכרון המוגדרות ב SDK.

בביצוע הניסויים העדפנו שלא להשתמש באלגוריתם RRT* מכיוון שהשימוש בו בסביבה הכוללת מכשולים יוצר מסלולים לא צפויים (מבחינת תוואי מדויק ומספר הפניות), ועבור סביבה ללא מכשולים השימוש בו היה שקול למסלול לינארי פשוט. ביצוע הניסויים באמצעות בחירה ידנית של נקודות מאפשר לנו לבדוק מגוון רחב יותר של תרחישים ולאתגר את הרחפן תוך התאמות לסביבה המוגבלת.

סיכום ועבודה להמשך

הראינו כי ישנה היתכנות לשימוש במספר רחפנים לביצוע משותף של משלוחים. ביצוע הפרויקט הותאם להטסה בתנאי מעבדה, אך ניתן להרחיב את אותם אלגוריתמים ושיטות לשימוש בעולם האמיתי.

עבור ניסויים הכוללים את המצלמה החיצונית, אפשרי להמשיך ולשכלל את אלגוריתם הזיהוי כך שיקלוט מכשולים בסביבה וימפה אותם בצורה אוטומטית. דבר זה יאפשר מגוון רחב יותר של ניסויים בתנאים מאתגרים יותר והתגברות על מכשולים בסביבה, בין אם סטטים או דינמיים. ישנן מספר רב של עבודות בתחום Image segmentation שיתאימו למשימה זו.

בהמשך כדאי לבצע ניסויים בסביבה חיצונית ובתנאים אמיתיים. במהלך עבודתנו הוגבלנו לניסויים בחדר סגור ומבוקר. סביבה זו נבדלת בעיקר בחוסר של הפרעות חיצוניות כמו רוח ומזג אוויר. בהתאם, יש לפתח את מנגנון הבקרה עבור שמירה על מסלול כך שידע להתמודד עם הפרעות יותר חזקות ובאופן יותר אמין. כיוון בסיסי הוא שימוש בבקר PID שיעזור לשמירה על מרחק יחסי בין הרחפנים. אופציה מורכבת יותר היא שימוש באלגוריתם MPC (model predictive control) שהראה תוצאות טובות עבור משימות כמו שמירה על מסלול מוגדר באמצעות רחפנים.

עבור מימוש עמיד מספיק לתנאי הסביבה בעולם האמיתי יש להעזר במצלמה שנמצאת על גבי הרחפנים. אפשרות אחת היא התגברות על חסמי השימוש של טלו באמצעות כרטיסי ארדואינו. אפשרות אחרת היא שימוש בסוג אחר של רחפנים שאינם מוגבלים באותה צורה. דגמים מתקדמים יותר מאפשרים בנוסף טווח פעולה גדול יותר משמעותית ויכולות הרמה משופרות.

רשימת מקורות

- (1) DJI Tello SDK
<https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf>
- (2) <https://github.com/dji-sdk/Tello-Python>
- (3) DJI TelloPy open-source library
<https://djitelopy.readthedocs.io/en/latest/>
- (4) YOLOv5
<https://github.com/ultralytics/yolov5>
- (5) YOLOv5 Pre trained weights for drone recognition
https://github.com/tusharsarkar3/Detect_Drone/

