

המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ו

סנכרון וחילוף רחפנים בזמן אמת וזיהוי והימנעות ממכשולים

Drones Synchronization and Obstacles

Avoidance

בן נקש

עדי גרינשפן

מנחה אקדמי:	מר שי תבור	אישור:	תאריך:
אחראי תעשייתי:	מר טל יצחק	אישור:	תאריך:
רכז הפרויקטים:	ד"ר אללוף מרים,		
	וד"ר יגל ראובן	אישור:	תאריך:

#	מערכת	
1	מאגר קוד	https://github.com/adiGr/FinalProjectDrone
2	יומן	https://trello.com/b/AuUIDKir/obstacles-detection-avoidance-project
3	ניהול פרויקט	Github

1. מבוא

חברת הסטארט-אפ Airscort מפתחת אפליקציות פורצות דרך בתחום טכנולוגיית הרחפנים בתעשייה. החברה הוקמה ע"י קבוצת מהנדסי מכונות, תוכנה וחשמל שרואים עתיד לטכנולוגיית הרחפנים.

הטכנולוגיה אותה מפתחים ב-Airscort תאפשר מגוון רחב של יישומי רחפנים אזרחיים, ביניהם:

- ❖ ספורט אתגרי - צילום הספורטאי בזמן אמת בצורה קלה, נוחה וללא צורך בגורם נוסף לניווט הרחפן.
- ❖ אבטחה – ביצוע סיורים, התבוננות מלמעלה וסיוע באבטחה שוטפת.
- ❖ בידור – ביצוע מגוון פעילויות מהנות בעזרת הרחפן כמו סיורים.
- ❖ משלוחים – משלוח מהיר ופשוט של מוצרים.
- ❖ חקלאות - ביצוע מעקב אחר גידולים חקלאיים ומתן מידע שימושי למשתמש.
- ❖ התאמה אישית – לצורך האישי של בעל עסק.

בעבר, רחפנים שימשו בעיקר לצרכים צבאיים, אך בשנים האחרונות גוברת מגמת השימוש בהם, וישנו ביקוש רב יותר בתעשייה האזרחית ובתעשייה העסקית.

חברת Airscort מעוניינת שהרחפן יהיה מסוגל לטוס בצורה אוטומטית ללא התערבות אדם, ולכן עליו לזהות מכשולים הנקרים בדרכו ולהימנע מהם.

בעיה נוספת העומדת בפני חברת Airscort היא שכיום זמן סוללת הרחפנים הוא נמוך מאוד (בין 15-20 דקות) ולכן, החברה מתמקדת בפיתוח מערכות מתקדמות אשר יאריכו את חווית המשתמש.

אחת הדרכים להארכת חווית המשתמש היא יכולת זיהוי של הרחפן לדעת כי סוללתו חלשה, ולבצע חילוף בינו לבין רחפן אחר באופן אוטומטי.

תיאור הבעיה

נרצה שרחפן יהיה מסוגל להחליף את האדם בביצוע פעולות שונות (או לאפשר פעולות שהאדם לבדו אינו מסוגל לבצע), ולאפשר לאדם להתפנות לעיסוקים האחרים. לדוגמה, החקלאי נדרש להגיע לשדה שלו ולבדוק מה מצבו באופן אישי – דבר המבזבז זמן רב. לכן, נדרוש שהרחפן יעבוד באופן אוטונומי וללא התערבות של גורם חיצוני. בצורה זו, אותו אדם יוכל לשלוח את הרחפן לבדוק את מצב השדה החקלאי, כאשר בו בזמן יוכל להמשיך בעיסוקיו השונים.

דוגמה נוספת הינה כאשר אדם מבצע ספורט אתגרי ומעוניין לתעד את עצמו. כיום, אדם אינו יכול לעשות זאת תוך כדי הפעילות בה הוא עוסק ונדרש אדם נוסף שיצלם אותו. לכן, נרצה שהרחפן יתעד את האדם ללא צורך בגורם נוסף שישלט בו.

לכן, הפרויקט שלנו מתחלק ל-2 בעיות אותן נרצה לפתור:

- ❖ שיפור יכולת הטיסה האוטונומית של הרחפן ע"י זיהוי והימנעות ממכשולים הנקרים בדרכו בזמן אמת באמצעות חיישנים שיורכבו על הרחפן.
- ❖ ביצוע חילוף בין שני רחפנים ע"י תיאום וסנכרון מידע על מנת למנוע מצב של התנגשות ביניהם (בעת החילוף, נתונים שני רחפנים אשר עוקבים אחרי אותה נקודת ציון, ולכן ישנה סכנת התנגשות).

דרישות ואפיון הבעיה

נתבונן כעת מנקודת המבט של אדם אשר ישתמש ברחפן ובו המערכת שאנו מפתחים. הדרישות שלנו מן הרחפן והתוכנה יהיו כדלקמן:

זיהוי והימנעות ממכשולים:

- ❖ לא נרצה שיהיה צורך לכוון את הרחפן מחדש בכל פעם שיש מכשול כלשהו בנתיבו (כיול).
- ❖ הרחפן ידע להימנע בצורה יעילה וטובה ממכשולים הנקרים בדרכו תוך כדי המשך עקיבה אחר המטרה המצולמת.

חילוף רחפנים:

- ❖ החילוף בין שני רחפנים יתבצע מהר עד כמה שניתן וזאת מ-2 סיבות:
 - מתבזבז זמן סוללה לשני הרחפנים.
 - נרצה שהצילום יהיה רציף והמשכי ללא איבוד מידע.
- ❖ נדרוש שהחילוף יתבצע באופן בטיחותי ויעיל. כלומר, לא נרצה שהחילוף ידרוש מרחב תמרון גדול מידי.
- ❖ נדרוש שהחילוף בין הרחפנים יתבצע בצורה כזו שלא יהיה חשש לפגיעה באדם אחר, כלומר שהחילוף יתבצע גבוה מעל הקרקע.

הבעיה מבחינת הנדסת תוכנה

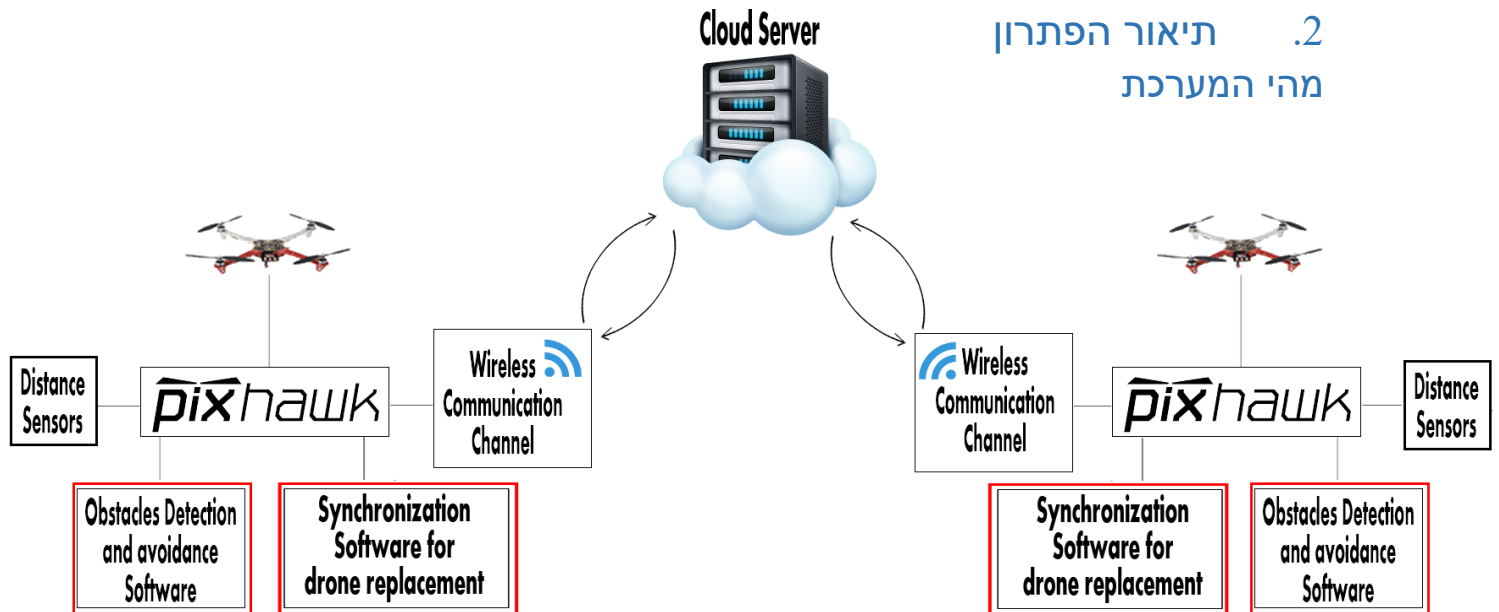
בפרויקט זה נעבוד מול בקר הטיסה של הרחפן מסוג Pixhawk. בקר זה מתבסס על סביבת קוד פתוח, ולכן נתכנת בשפות C++ ו-C. ישנן מספר בעיות תכנותיות אשר אנו נאלצים לפתור:
א. שאיבת נתונים בזמן אמת לצורך זיהוי מכשולים מחיישני המרחק שיורכבו על גבי הרחפן ויחוברו לבקר ה-Pixhawk.

ב. הימנעות מפגיעה במכשול אשר התגלה בדרכו של הרחפן – נצטרך לפתח מערכת תוכנה אשר תדע לנווט את הרחפן לנתיב בטוח בצורה כזו שלא יפגע במכשול אשר זוהה בדרכו או במכשול אחר שעשוי להיות בנתיב המילוט.
בשוק כבר קיימים אלגוריתמים לזיהוי מכשולים, אך הם עושים שימוש בטכנולוגיה מתקדמת ויקרה שאינה במסגרת התקציב שלנו. לכן, אנו צריכים לפתח ולהתאים אלגוריתם אשר יאפשר את זיהוי מכשולים במסגרת התקציב והחומרה המוגבלים שברשותנו.

ג. סנכרון רחפנים – בעת שרחפן מזהה כי סוללתו חלשה, הוא ישלח בקשה לחילוף. כאן אנו נדרשים לפתח מערכת תוכנה אשר תדע לסנכרן בין 2 הרחפנים בצורה כזו שהחילוף יתבצע באופן:

- a. **מהיר:** על מנת לחסוך בזמן סוללה לשני הרחפנים.
- b. **יעיל:** גם עבור חיסכון בזמן סוללה וגם כיוון שלא נרצה שהחילוף ידרוש מרחב תמרון גדול מידי.
- c. **רציף:** כיוון שמטרת הרחפן היא לעקוב באופן רציף אחרי המטרה (גולש הסקי במקרה שלנו), נרצה שהחילוף בין הרחפנים לא ישפיע על המעקב ועל רצף הצילום. לכן, האלגוריתם שנפתח יהיה חייב לבצע את 'חילוף המשמרות' בצורה כזו שלא תגרום לאיבוד מידע.

2. תיאור הפתרון מהי המערכת



התרשים הקודם מתאר את ארכיטקטורת המערכת ומורכב מכמה חלקים:

❖ **בקר ה-Pixhawk** הינה מערכת טייס אוטומטי המבוססת על קוד פתוח ופותחה ע"י חברת 3D Robotics. בבקר זה משולבים טכנולוגיית חיישנים ומעבד מתקדמים, מערכת הפעלה בזמן אמת המספקת ביצועים מרשימים, גמישות ושליטה לכלים אוטונומיים. דרך בקר זה נשאב את הנתונים מן החיישנים, ואילו נצרב את האלגוריתמים שנפתח (הפירוט עליהם בסעיפים הבאים).

❖ **חיישני מרחק** – חיישני מרחק יורכבו על הרחפן ויסרקו את השטח שמולם באופן תמידי. נתוני הסריקה ישלחו אל בקר ה-Pixhawk בזמן אמת וינותחו ע"י האלגוריתמים שנפתח ונצרב אל השבב שלו. בפרויקט זה בחרנו להשתמש בחיישנים העובדים על לייזר, וזאת כיוון שהם מאפשרים סריקה לטווח רחוק יותר.

❖ **תוכנה לזיהוי והימנעות ממכשולים** – תוכנה אשר נפתח ונצרב לבקר ה-Pixhawk. זו תעבוד על הבקר ותקבל כקלט את נתוני המרחק מהחיישנים, ותשלח לרחפן פקודות תזוזה להימנעות מהמכשולים שזוהו.

❖ **תוכנה לסנכרון הרחפנים** - תוכנה אשר נפתח ונצורב לבקר ה-Pixhawk הנמצא בכל אחד מהרחפנים. התוכנה תיעזר בנתוני מיקום של האדם המצולם אשר ישלחו אליו משרת הענן על מנת להמשיך את הפעולה המתבצעת. נכון לרגע זה, איננו רואים צורך ממשי לשליחת נתוני המיקום של הרחפן המוחלף למחליפו (דרך השרת) כיוון שאנו יוצאים מנקודת הנחה כי גובה העקיבה של הרחפן הינו קבוע, והעברת מידע נוסף יצור עוד עומס על הרשת ועל המעבד בבקר ה-Pixhawk.

תיאור הפתרון המוצע:

זיהוי והימנעות ממכשולים:

- א. חיבור החיישנים לבקר ה-Pixhawk באמצעות חיבור I2C, בדיקת החיישנים לתקינות קלט ע"י הזזת חפצים מול הרחפן.
- ב. פיתוח האלגוריתם ראשוני של זיהוי המכשולים והבנת מיקומם ביחס לרחפן.
- ג. סימולציות ראשוניות, הרצת האלגוריתם ובדיקתו באמצעות הדפסות למסך.
- ד. פיתוח פקודות תזוזה לרחפן בהתאם למיקום המכשול שזיהינו באמצעות האלגוריתם שפותח ונבדק בשלב קודם.
- ה. סימולציות, והרצות נוספות. בדיקת האלגוריתם בזמן אמת תוך כדי הפעלת הרחפן.

סנכרון רחפנים באוויר בזמן אמת:

- א. בדיקת ערוץ השידור בין הרחפן לשרת הענן ומדידת זמני העברת הנתונים לתקינות.
- ב. פיתוח אלגוריתם ראשוני להזזת הרחפן אשר סוללתו חלשה בצורה שתאפשר לרחפן המחליף להחליפו ולהימנע מהתנגשות.
- ג. סימולציה ראשונית לבדיקת תקינות האלגוריתם ע"י הדפסת נתוני המיקום של כל אחד מהרחפנים תוך כדי החילוף ביניהם.
- ד. שיפור האלגוריתם, תיקון באגים. הרצת סימולציות נוספות וביצוע הרצה מלאה.

תיאור הכלים המשמשים לפתרון

❖ **Mission Planner**

בעזרת תוכנה זו נוכל לבצע מגוון פעולות עם הרחפן, ביניהן:

1. מתן נקודות ציון ומסלול מעוף לרחפן בעזרת Google Maps (תכנון משימות טיסה).
2. הורדת קבצי Log של המשימה וניתוחם.

❖ **Arduino Software (IDE)**

סביבת העבודה של Arduino. נוכל להיעזר בה על מנת לשאוב את הנתונים הנמדדים מהחיישנים וביצוע ניסויים שונים בקלות יחסית (ישמש אותנו בעיקר בתחילת העבודה עם החיישנים וצבירת ניסיון, יותר מאוחר נעבור לתכנות בקר ה-Pixhawk).

❖ **ArduPilotMega (APM)**

פלטפורמה מבוססת קוד פתוח של כלי טיס בלתי מאוישים שבעזרתה ניתן לשלוט על רחפנים אוטונומיים. בעזרת Eclipse נוכל לערוך את קוד ה-ArduPilot שיתאים לצרכים שלנו ולתת לבקר ה-Pixhawk מטרות.

❖ **DroneKit-Python API**

API המאפשר למפתחים ליצור תוכניות שירוצו על המעבד הנמצא ברחפן. אותו מעבד מדבר ושולט על ה-Autopilot. תוכניות הרצות על מעבד זה מסוגלות לבצע משימות עיבוד אינטנסיביות ומשימות תלויות זמן אמת כמו זיהוי מכשולים ושינוי נתיב טיסה. API זה עובד בסביבות העבודה Unix ו-Windows ונותן לנו (המשתמשים) גישה לביצוע משימות כמו:

1. התחברות לרחפן (או מספר רחפנים) ע"י סקריפט.
2. שינוי וקבלת מצב הרחפן ונתונים שונים לגביו.
3. קבלת מסרים אסינכרוניים על שינויי מצבים.
4. הדרכת הרחפן לטוס למיקום מסוים.
5. שליחת פקודות שרירותיות לשליטה על תנועת הרחפן וחומרה נוספת במידה ויש.

סקירת עבודות דומות בספרות והשוואה

- ❖ Autonomous quadcopter, obstacle detection on Arduino
<https://github.com/bitcoinsoftware/UrbanDroneSystem>
קוד לזיהוי מכשולים של רחפן הכתוב ב-Arduino.
- ❖ source code used to control a Quadcopter
<https://github.com/2012SEng466/copter>
מאגר זה מכיל את קוד מקור משמש לשליטה ברחפן מכיל תכנית ב-Arduino לקריאת פקודות סידורי ממחשב ולשלוח פקודות מוטוריות לרחפן.
- ❖ Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadcopter.
[http://www.informatik.uni-wuerzburg.de/fileadmin/10030800/user_upload/quadcopter/Paper/Gageik_Mueller_Montenegro_2012_OBSTACLE_DETECTION_AND COLLISION_AVOIDANCE_USING ULTRASONIC DISTANCE SENSORS FOR AN AUTONOMOUS QUADROCOPTER.pdf](http://www.informatik.uni-wuerzburg.de/fileadmin/10030800/user_upload/quadcopter/Paper/Gageik_Mueller_Montenegro_2012_OBSTACLE_DETECTION_AND_COLLISION_AVOIDANCE_USING_ULTRASONIC_DISTANCE_SENSORS_FOR_AN_AUTONOMOUS_QUADROCOPTER.pdf)
מאמר על זיהוי מכשולים והימנעות התנגשות בעזרת חיישני אולטרסאונד.
- ❖ PIXHACK
<http://copter.ardupilot.com/wiki/common-pixhawk-wiring-and-quick-start/>
הסבר על איך ניתן לשלוט על ה-Pixhawk. והסבר על הבקר.

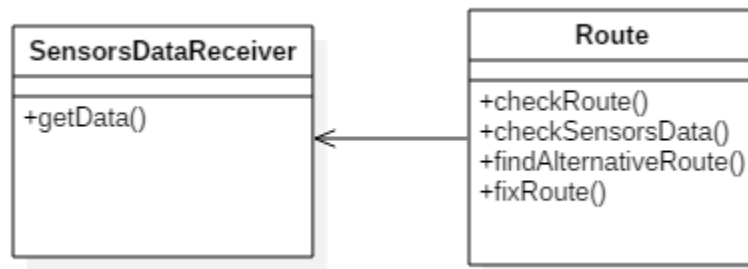
נספחים

א. ביבליוגרפיה

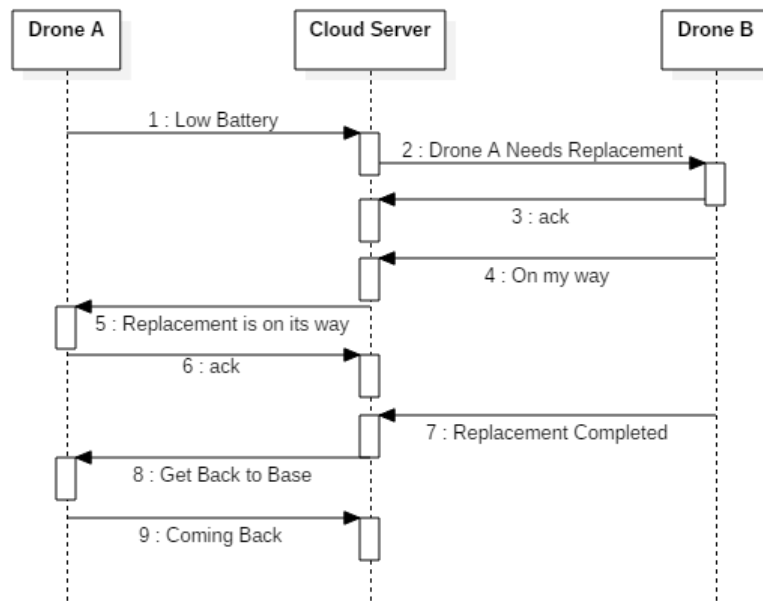
- ❖ <http://copter.ardupilot.com/>
- ❖ <https://phidgets.wordpress.com/2014/05/23/exploring-the-many-methods-of-object-detection/>
- ❖ <http://diydrones.com/>

ב. תרשימים וטבלאות

- ❖ Class Diagram כללית של התוכנה להימנעות ממכשולים וניתוב מחדש של הרחפן:



- ❖ Sequence Diagram של חילוף בין שני רחפנים:



ג. תכנון הפרויקט

היכרות ותחילת עבודה עם Aircort	04.10.15
התחלת ביצוע מחקר על חיישנים	01.11.15
הגשת הצעת פרויקט	22.11.15
רכישת חיישן מרחק ראשון ולמידת סביבת העבודה	01.12.15
הגשת אב טיפוס - ביצוע ניסויים ובדיקות ראשונות לחיישן המרחק	17.01.16
מתן פקודות לרחפן דרך בקר ה pixhawk לתזוזתו. סימולציות של הרחפן לתגובת הפקודות. בדיקה ומדידת זמני העברת הנתונים בערוץ השידור בין הרחפן לשרת הענן.	01.03.16
חיבור חיישנים נוספים לבקר ה pixhawk. בדיקה והמשך פיתוח אלגוריתם להימנעות ממכשולים. מימוש אלגוריתם להזזת הרחפן המוחלף מן הרחפן המחליף על מנת להימנע מהתנגשות.	01.04.16
בניה – הרצת סימולציות ראשוניות על תוכנת הימנעות ממכשולים ובדיקת התוכנה להימנעות ממכשולים. בדיקת תקינות האלגוריתם ע"י הדפסת נתוני מיקום של כל אחד הרחפנים תוך כדי חילוף.	05.05.16
בדיקות התוכנה להימנעות ממכשולים תוך כדי הפעלת הרחפן. שיפור אלגוריתם להחלפת הרחפנים והרצת סימולציות נוספות בזמן אמת.	1.06.16
מסירה	19.06.16
העברה.	07.07.16

ד. טבלת סיכונים

#	הסיכון	חומרה	מענה אפשרי
1	אי עמידה בזמנים	5/5	הוספת שעות נוספות
2	תלות בצוות מקביל	3/5	סנכרון, פגישה ותיאום ציפיות עם הצוות המקביל.
3	אי זמינות של הרחפן לסימולציה	2/5	קביעת זמן שימוש ברחפן מספיק זמן מראש
4	חיבור החיישן לא מתאים לבקר ה-Pixhawk	2/5	מחקר מראש על החיישן, חיבוריו ואופן פעולתו.
5	בעיה בהוספת החיישנים לתוכנה בעזרתה נבצע את הסימולציות	4/5	קריאה מרובה של החומר וצפייה בקודים ב-github עבור בקר ה-arduino. בקשת עזרה מחברי הצוות היותר מנוסים עם תוכנה זו.
6	חוסר ידע באמולטור המתאים לרחפן וסביבות העבודה שלו.	2/5	למידת האמולטור וסביבות העבודה של הרחפן.
7	תכנות לא יעיל שעלול לגרום לעומס חריג על המעבד ולגרום בנוסף לבזבז סוללה.	5/5	ניתוח זמן הריצה של הקוד שכתבנו וכתיבתו באופן יעיל עד כמה שנוכל.
8	חוסר ידע בבקר ה-Pixhawk ואופן פעולתו עם הרחפן.	4/5	למידת הפקודות שדרך הבקר מתקשר עם חומרת הרחפן, אופן העברתן וביצוען.
9	עיכוב בפיתוח צד השרת וקבלת המידע ממנו	4/5	תלות בצוות אחר אשר מפתח את צד השרת. ניתן נקודות ציון משלנו במקום לביצוע הניסויים.
10	פגיעה חמורה ברחפן	3/5	הרצת בדיקות וסימולציות יבשות לפני ביצוע ניסויים חיים עם הרחפן.
11	עבודה חופפת בתוך הצוות	2/5	משימות מוגדרות ושונות לחברי הצוות ליעול העבודה בפרויקט
12	אי הגעה של החיישנים בזמן	4/5	כתיבת קוד עם חיישנים אחרים וניסיון להתאים לחיישנים בזמן הגעתם.

ה.טבלת דרישות

#	תיאור
1	מחקר על חיישנים: 1. מהם סוגי חיישני המרחק הקיימים. 2. יתרונות וחסרונות של כל אחד מן הסוגים הקיימים. 3. קבלת החלטה בהתאם לאילוצים שונים: a. משקל. b. צריכת חשמל. c. מחיר. d. טווח פעולה. e. זווית פעולה.
2	פיתוח אלגוריתם לזיהוי מכשולים: 1. קליטת נתוני מדידה של החיישנים דרך בקר ה-Pixhawk. 2. ניתוח הנתונים, ומתן התראה במידה וזוהה מכשול.
3	פיתוח אלגוריתם להימנעות ממכשולים: 1. בדיקה לקבלת התראת מכשול. 2. פיתוח מספר גישות שונות להימנעות מן המכשול (עקיפה מימין/משמאל/מלמעלה ועצירה לצורך בדיקות). 3. מתן נתיב אלטרנטיבי ושינוי נתיב הטיסה.
4	פיתוח אלגוריתם להחלפת רחפנים באופן יעיל: 1. בדיקת מצב סוללה. 2. בדיקת מרחק מבסיס טעינה קרוב. 3. חישוב צריכת סוללה הצפויה עד להגעת הרחפן המחליף. 4. חישוב צריכת סוללה הצפויה עד להגעה לבסיס. 5. שליחת בקשת חילוף לשרת בעת שזוהתה סוללה חלשה ובהתאם למרחק מן הבסיס הקרוב. 6. קבלת מסר מהשרת להחלפת רחפן אחר. 7. ביצוע החילוף ע"י העלאת הרחפן המוחלף גבוה יותר על מנת לאפשר למחליפו להגיע ללא סכנת התנגשות.
5	הרצות חיות ועשיית מבדקים לכל אורך הפיתוח: 1. בדיקת הקלט מהחיישנים. 2. בדיקת האלגוריתם לזיהוי מכשולים. 3. בדיקת האלגוריתם להימנעות ממכשולים. 4. הרצה חיה של הרחפן לבדיקת האלגוריתם להימנעות ממכשולים. 5. בדיקת אלגוריתם להחלפת רחפנים בעזרת הדפסות למסך של מיקום הרחפנים בכל זמן נתון לאחר בקשת חילוף. 6. הרצה חיה ובדיקת האלגוריתם להחלפת הרחפנים.