

חוות דעת מנחה

שם הפרויקט – רחפן חצי אוטונומי

בפרויקט זה מפותח רחפן להפעלה בתוך מנהרות. הרחפן יכלול שני מצבי טיסה, אחד ידני לשליטה מרחוק כאשר טווח התקשורת מאפשר קשר עם המפעיל ושני אוטונומי על בסיס חיישנים. גם בשלב הידני יופעלו אמצעי עזר למטיס. הרחפן מתוכנן ונבנה במסגרת הפרויקט ומתבסס על מודולים קיימים (כגון מסגרת קרבון, בקר-טיסה מנועים ועוד). תכנון כזה דורש ידע רחב בתחום הרחפנים וניסיון בהפעלת הציוד הרלוונטי. במהלך הסמסטר האחרון הסטודנט בנה את הרחפן הדרוש וביצע ניסויי טיסה רבים כדי ליישם בהתאם למטרות הפרויקט. פיתוח המערכת דורש גם חיבור למחשב טיסה בו ימומשו אלגוריתמים רלוונטיים אותם יפתח הסטודנט. בשלב הנוכחי הסטודנט ביצע את החיבורים והכילים הרלוונטיים והתנסה בפיתוח קוד ראשוני. האלגוריתמים הנחוצים הינם חוגי בקרה והנחיה שיפעלו במקביל לחוגי הבקרה של הרחפן כדי להקל על תהליך ההטסה בטווח הצפוף של המנהרה וכדי להתמודד עם בעיה של אובדן תקשורת. פיתוחם יתבסס על המודל הדינמי של הרחפן ומחייב הבנה טובה של עקרונות הטיסה והבקרה של הרחפן. בסמסטר הנוכחי, מלבד הבניה והכילול של הרחפן, הסטודנט למד את הרקע התיאורטי הנדרש להמשך הפרויקט.

במהלך הסמסטר בוצעה עבודה מצוינת בהיקף מתאים לחלק הראשון של הפרויקט. אתגרים רבים קיבלו מענה ונסללה הדרך להמשך העבודה הנדרשת להשלמת הפרויקט.

מנחה: שי ארוגטי

תאריך: 24/05/25



חתימה:



סאוניברסיטת בן-גוריון בנגב
הפקולטה למדעי ההנדסה
המחלקה להנדסת מכונות



דו"ח פרויקט

רחפן חצי אוטונומי

24-102

שם הסטודנט: עמית גדג' 207653965

תאריך הגשה: 27.05.2024

שם המנחה: שי ארוגטי



Ben Gurion University
Faculty of Science Engineering
Mechanical Engineering Department



Final Project Report

Half Autonomous Drone

24-102

Author: Amit Gedj 207653965

Date: 27/05/2024

Supervisor: Shai Arogeti

May 2024

Iyar 5784

תקציר

במסגרת מלחמת חרבות ברזל עלה הצורך באמצעי המאפשר סריקה של מנהרות ללא סיכון של חיי אדם. כיום, ישנם מספר רחפנים בשוק הנותנים מענה לצורך זה ופרויקט זה מתחרה ברחפנים אלו ואף מציע פונקציות נוספות. פרויקט זה מתמקד בבניית רחפן המסוגל לטוס על ידי מפעיל ובנוסף לטוס בצורה אוטונומית. נקודת המעבר בין שתי אפשרויות ההטסה הינה נקודת איבוד התקשורת של המפעיל עם הרחפן. בנקודת איבוד התקשורת הרחפן ימשיך טיסה וסריקה במנהרות עד הנקודה בה הסוללה לא תאפשר לו תעופה מלאה חזרה אל נקודת התקשורת האחרונה עם המפעיל. יכולת הטיסה האוטונומית מאפשרת חקירה מעמיקה יותר של המנהרות, דבר אשר כיום מוגבל על ידי טווח הקליטה של המפעיל עם הרחפן.

עד כה, הפיתוח עסק בבניית פלטפורמה ניסויית המאפשרת הטסה נוחה ובעלת מערכות עזר המאפשרות שמירה על מיקום הרחפן במידה ולא מתקבלות פקודות מהמפעיל. על בסיס פלטפורמה זו ייכתבו התוכניות הרלוונטיות לפרויקט ובדיקת היתכנות של הרעיון. הפלטפורמה מורכבת ממספר רכיבים בסיסיים אשר חיבורם ביחד מאפשר את הטיסה הבסיסית של הרחפן- שלדה, בקר טיסה, חיישן *Optical Flow*, מנועים ופרופלורים. מערכת ההטסה של הרחפן מבוססת על תוכנה קיימת בשם *ArduPilot*, תוכנה זו מבצעת את כלל חוגי הבקרה והשיערוך הנדרשים לטובת הטסת הרחפן ושליטה בחיישנים נוספים.

על מנת לאפשר את ההטסה האוטונומית, על הפלטפורמה מורכבים רכיבים נוספים- חיישני *LiDAR* ומחשב זעיר. שימוש בתוכנה זו מאפשר התמקדות בחלק החדשני בפרויקט זה. המחשב מקבל את הנתונים מחיישני ה-*LiDAR*, מבצע עיבוד נתונים ושולח לבקר טיסה פקודות הטסה. כלל הרכיבים המוזכרים מעלה חוברו למערכת אחת המאפשרת את בחינת הרעיון המוצג בפרויקט.

לאחר הרכבת הרכיבים הבסיסיים של הרחפן, ולאחר תהליך כיוולים של אותות הבקרה, יחד עם מספר טיסות מבחן הבודקות את תגובת הרחפן, התקבלו מספר טיסות מוצלחות המראות כי בקר הטיסה, יחד עם חיישן ה-*Optical Flow* מאפשרים הטסה נוחה ושהייה של הרחפן במקום. דברים בסיסיים הנדרשים לצורך המשך הפרויקט ותחילת פיתוח האלגוריתמים האוטונומיים.

Abstract

As part of the Iron Sword War, the need arose for a means of scanning tunnels without endangering human life. Today, there are a few drones on the market that meet this need and this project competes with these drones and even offers additional functions. This project focuses on building a drone capable of flying by an operator and flying autonomously. The crossing point between the two flight options is the operator's point of loss of communication with the drone. At the point of loss of communication, the drone will continue flying and scanning the tunnels to the point where the battery will not allow it to fully fly back to the last point of communication with the operator. The autonomous flight option enables a deeper investigation of the tunnels, which is currently limited by the operator's range with the drone.

So far, the development has involved building an experimental platform that enables comfortable flight and has auxiliary systems that allow maintaining the drone's position if no commands are received from the operator. Based on this platform, the relevant plans for the project and a feasibility study of the idea will be written. The platform consists of several basic components whose connection together enables the drone's basic flight – chassis, flight controller, optical flow sensor, engines, and propellers. The drone's flight system is based on existing software called ArduPilot, which performs all the control and evaluation circles required to fly the drone and control additional sensors .

In order to enable autonomous flight, additional components are assembled on the platform – LiDAR sensors and a miniature computer. Using this software allows focusing on the innovative part of this project. The computer receives the data from the LiDAR sensors, performs data processing, and sends flight commands to the flight controller. All the components mentioned above were connected to a single system that enables the examination of the idea presented in the project .

After assembling the basic components of the drone, and after a process of calibrating the control signals, together with several test flights that test the drone's response, several successful flights were obtained showing that the flight controller, together with the Optical Flow sensor, enable comfortable flight and stay of the drone in place. Basic things required for the continuation of the project and the beginning of the development of autonomous algorithms.

תוכן עניינים

1	מבוא	1
2	רקע	2
2.1	כלי טיס מסוג רב-להב	2.1
2.2	מודל קינמטי	2.2
2.3	מודל דינמי	2.3
2.4	חיישנים	2.4
2.5	מנועים ללא מברשות	2.5
2.6	חיבורי תקשורת	2.6
2.7	סוללת Li – Po וסוללת Li – ion	2.7
2.8	תוכנות שליטה ברב-להב	2.8
2.9	ספריית DroneKit	2.9
3	דרישות המערכת	3
3.1	גודל ומשקל הרחפן	3.1
3.2	זמן תעופה	3.2
3.3	יכולות חישה	3.3
3.4	מערכות עזר	3.4
4	חלקי הפיתוח שבוצעו עד כה	4
4.1	הרכבת הרחפן	4.1
5	תכנית להמשך	5
5.1	הפעלת חיישני LiDAR	5.1
5.2	קבלת מידע מבקר הטיסה ומתן פקודות	5.2
5.3	תכנון מערכות בקרה וניווט	5.3
5.4	מימוש מערכות הבקרה והניווט	5.4
6	ביבליוגרפיה	6

רשימת איורים

- איור 1 - קונפיגורציות גוף שונות וכיוון סיבוב המנועים.....2
- איור 2 - מערכות הצירים המוגדרות עבור מודל זה.....3
- איור 3 - תיאור של החיישנים והמדידות השונות מחיישן IMU.....5
- איור 4 - אופן הפעולה של חיישן *LiDAR*.....6
- איור 5 - חיישן *Optical Flow* בסיסי בעל חיבור *UART*.....6
- איור 6 - סכמה של מנוע ללא מברשות.....6
- איור 7 - איור סכמטי של חיבור רכיבים על ידי חיבור מסוג *UART*.....7
- איור 8 - איור סכמטי של חיבור מספר רכיבים לבקר אחד.....7
- איור 9 - תוצאה רצויה מהשוואת *Optical Flow* ל- *Gyro*.....10
- איור 10 - חיישן *Optical Flow Micoair MTF – 01*.....10
- איור 11 - השוואה בין ערכי ה-*Gyro* לערכי החיישן *Optical Flow*.....10
- איור 12 - השוואה בין ערכים מכוילים.....11
- איור 13 - ספריות בהן יש שימוש בקוד פייתון.....12
- איור 14 - קטע קוד ראשי.....12
- איור 15 - קטע קוד ראשי- מחלקת *UAV*.....12
- איור 16 - פונקציות- מחלקת *UAV*.....13

רשימת טבלאות

טבלה 1 – רשימת רכיבים שנבחרו לאחר ניסויים.....11

רשימת סימנים

סימן	יחידות מידה	תיאור מילולי
ψ	rad	זווית סבסוב- עולם
θ	rad	זווית עלרוד- עולם
ϕ	rad	זווית גלגול- עולם
r	rad	מהירות סיבוב גלגול- גוף
p	rad	מהירות סיבוב עלרוד- גוף
q	rad	מהירות סיבוב סבסוב- גוף
ω	$\frac{rad}{sec}$	מהירות סיבוב מנוע
F	N	כוח דחף
τ	$N \cdot m$	מומנט

1. מבוא

כלי טיס בלתי מאוישים, או בשפה הצבאית המקובלת- כלי טייס מאוישים מרחוק (כטמ"מ), הינם רובוטים הלוקחים חלק משמעותי בחיי היום-יום של מסגרות רבות בעולם. גופים פרטיים משתמשים בכטמ"מים לטובת אבטחת משרדים, תחרויות הטעות, צילום וכו'. גופים ביטחוניים משתמשים בכטמ"מים לטובת לאבטח כוחות רגליים בשטח, סריקת מבנים ושטחים מוסתרים לעין כוחותינו. כאשר טווח השימושים כה רחב, קיימים נותני שירות רבים אשר נותנים מענה לתחומים נקודתיים ועל כן, לא לכל צורך קיים מענה. אחד מהתחומים אשר המענה בהם דל הינו רחפן אוטונומי במרחבים סגורים.

רוב הרחפנים אשר נמצאים כיום בשוק מיועדים לשימוש חיצוני, בשטחים פתוחים. היצרניות הגדולות של כלי הטיס, כגון DJI, משתמשות בחיישנים מתקדמים על מנת לאפשר יציבות של כלי הטיס. החיסרון העיקרי של חיישנים אלו הינו ההתבססות על תמונות וזיהוי שינוי חלקיקים במרחב. עבור תווך סגור אשר רווי באבק עקרון פעולה זה הינו בעייתי ולא נותן פתרון אופטימלי לייצוב הרחפן. כיום, מנהרות הן כלי בו נעשה שימוש בתחומים רבים ובין היתר בתחום הצבאי.

כתוצאה מכך שהמנהרות נמצאות מתחת לפני השטח, חקירת תוואי זה על ידי גורם אנושי מהווה סכנה ושהיה בתנאים מאתגרים בהם, בין היתר, תקשורת מוגבלת עם העולם החיצוני. עולם המנהור הינו עולם רווי מכשולים ואתגרים מכיוון שאין מבנה מוגדר של מנהרה. מנהרות שונות יכולות להיות בגדלים שונים, עם חסימות שונות ועם תשתיות שונות העוברות בהן. כמו כן, שהיה במנהרות עלולה לכלול תנאי סביבה קיצוניים כגון, מחסור בחמצן או טמפרטורות גבוהות. כתוצאה מהקשיים אשר הוצגו קודם לכן, ישנן סכנות רבות בתחום המהוות סכנת חיים ועקב כך שהתקשורת עם העולם החיצון הינה מוגבלת, אפשרויות החילוץ גם הן מוגבלות.

על מנת להתמודד עם בעיה זו ועם בעיות נוספות הקיימות בשדה הקרב, נבחנת האופציה של פיתוח רחפן המבצע חקירה וסקירה מוקדמת של המנהרות. רחפן זה מיועד לטוס בהפעלה ידנית על ידי מפעיל עד לנקודת איבוד התקשורת בין השניים ולאחר מכן הרחפן ימשיך לנוע בצורה אוטונומית, עד הנקודה הרחוקה ביותר שאליה הסוללה מאפשרת להגיע ולאפשר תנועה חזרה לנקודת התקשורת. תנועת הרחפן תתבסס על אמצעי

חישה שונים כגון- חיישני **LiDAR** ו- **Optical Flow**.

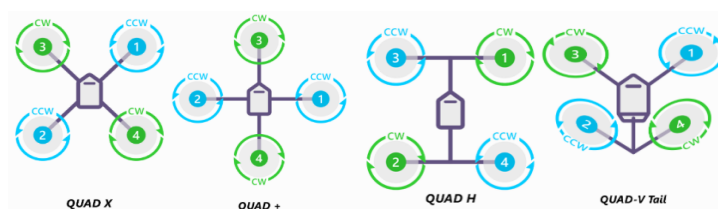
דו"ח זה יציג את הרקע הנדרש לפרויקט ויכלול מושגים בסיסיים הנדרשים לפרויקט- רחפנים והדינמיקה שלהם, חיישנים, סוגי תקשורת ועוד. בנוסף, בדו"ח יוצגו האתגרים בפרויקט ותוצג תוכנית עבודה להמשך.

2. רקע

פרק זה יעסוק ברקע התאורטי ובסקר ספרות המקנה את הידע הבסיסי הנדרש בפרויקט זה.

2.1. כלי טיס מסוג רב-להב

כלי טיס מסוג רב-להב הינו כלי טיס המייצר כוח עילוי על ידי סיבוב מדחף בעל להבים. כלי טיס מסוג זה הנפוץ ביותר הינו בעל ארבעה מדחפים הממוקמים בכל אחד מקצוות מסגרת כלי הטיס. מסגרת כלי הטיס יכולה להיות בנויה במספר תצורות שונות כפי שניתן לראות באיור 1. כמו כן, כל זוג מנועים נגדיים מסתובב בכיוון הפוך, זאת על מנת לבטל את המומנטים הפועלים על הגוף.



איור 1: קונפיגורציות גוף שונות וכיוון סיבוב המנועים

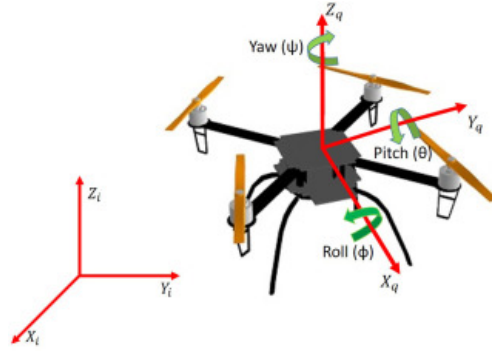
כאשר כלל המנועים מסתובבים באותה מהירות, כל מנוע מייצר את אותו כוח דחף. כאשר כוח הדחף הכולל של המנועים גדול מכוח הכבידה הפועל על כלי הטיס, תתקבל הרמה של הכלי. על מנת ליצור תנועה כיוונית של הכלי נדרש ליצור כוחות לא שווים בין המנועים באמצעות מהירויות סיבוב שונות. לדוגמא, עבור תצורת גוף של QUAD H, כאשר מנועים 1 ו-4 נעים מהר יותר ממנועים 2 ו-3, מתקבלת זווית גלגול וכתוצאה מכך כוח הדחף אשר מפעילים המנועים פועל בשני צירים שונים ומתקבלת תנועה צידית. כתוצאה מפירוק כוח הדחף למספר צירים, הכוח שפועל בכל ציר הינו קטן יותר מהכוח השקול ולכן יש צורך במהירות סיבוב גבוהה יותר של המנועים על מנת לייצור כוח דחף גדול מספיק המאפשר שהיה בגובה קבוע. היתרונות הבולטים של הרב-להב הינם יכולת לנשיאת משקל רב, יכולת טיסה במבנים סגורים או באזורים צפופים, זאת עקב האפשרות לרחפנים בגדלים קטנים מאוד, ובעל פשטות מכנית. כלי טיס זה משמש במספר רב של תחומים- ביטחוני, מרוצים, צילום, שליחויות ועוד.

2.2. מודל קינמטי

המודל הקינמטי מתאר את ההמרה של מהירויות כלי הטיס ממערכת הגוף למערכת העולם, כפי שהוצג על ידי אלון דויד, 2011. הפיתוח מתבסס על מספר הנחות:

- מרכז המסה של הכלי הינו במרכז השטח.
- הכלי הינו גוף קשיח.
- המנועים הינם קשיחים ומהווים חלק מהכלי.

המודל מתבסס על שתי מערכות צירים עיקריות. האחת הינה מערכת צמודת גוף והשניה מערכת העולם, כפי שמוצג באיור 2.



איור 2: מערכות הצירים המוגדרות עבור מודל זה. מערכת הגוף עם אינדקס q ומערכת העולם עם אינדקס i

לכלי ישנם שש דרגות חופש אשר כוללות את מצבו הזוויתי של הכלי במרחב ואת מיקום הכלי במרחב. על מנת להמיר את המהירות הזוויתית של הכלי מהמערכת הצמודת גוף למערכת העולם נדרש להשתמש במטריצת סיבוב המביעה את סיבוב הכלי סביב כל צירו.

$$R = \begin{bmatrix} c(\psi) c(\theta) & c(\psi) s(\theta) s(\phi) - s(\psi) c(\phi) & c(\psi) s(\theta) c(\phi) + s(\psi) s(\phi) \\ s(\psi) c(\theta) & s(\psi) s(\theta) s(\phi) + s(\psi) c(\phi) & s(\psi) s(\theta) c(\phi) - c(\psi) s(\phi) \\ -s(\theta) & s(\phi) c(\theta) & c(\phi) c(\theta) \end{bmatrix} \quad (2.1)$$

כאשר $\cos \triangleq c$; $\sin \triangleq s$

הזוויות מתארות סיבוב באופן הבא: זווית סבסוב ψ (yaw), זווית גלגול ϕ (roll) וזווית עלרוד θ (pitch) את המהירות הקווית של הגוף במערכת האינרציאלית ניתן לקבל על ידי הכפלת וקטור המהירויות הקוויות לפי מערכת הגוף במטריצת הסיבוב:

$$\begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{bmatrix} = \begin{bmatrix} c(\psi) c(\theta) & c(\psi) s(\theta) s(\phi) - s(\psi) c(\phi) & c(\psi) s(\theta) c(\phi) + s(\psi) s(\phi) \\ s(\psi) c(\theta) & s(\psi) s(\theta) s(\phi) + s(\psi) c(\phi) & s(\psi) s(\theta) c(\phi) - c(\psi) s(\phi) \\ -s(\theta) & s(\phi) c(\theta) & c(\phi) c(\theta) \end{bmatrix} \begin{bmatrix} \dot{X}_q \\ \dot{Y}_q \\ \dot{Z}_q \end{bmatrix} \quad (2.3)$$

את המצב הזוויתי של הכלי במערכת אינרציאלית נציג על ידי זוויות אוילר כאשר במערכת הגוף המהירות הזוויתית של הכלי מובעת על ידי (p, q, r) . את הקשר בין המהירות הזוויתית של הכלי לבין ווקטור הנגזרות של זוויות אוילר ניתן להציג באופן הבא:

$$\begin{bmatrix} r \\ p \\ q \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s(\phi) \\ 0 & c(\phi) & c(\theta) s(\phi) \\ 0 & -s(\phi) & c(\phi) c(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.4)$$

2.3. מודל דינמי

המודל הדינמי מבוסס על הצגה בה כוח הדחף והמומנט של כל מנוע הינם פרופורציונליים למהירות הסיבוב של המנוע.

$$F = b\bar{\omega}^2 \quad (2.5)$$

$$\tau = k\bar{\omega}^2 \quad (2.6)$$

כאשר b ו- k הינם מקדמים המשתנים כתלות במספר גורמים, אך בעבודה זו ההנחה היא כי הינם קבועים וזהים בין המנועים. המשתנה $\bar{\omega}$ מתאר את המהירות הזוויתית של המנוע. כעת, ניתן להגיד כי כוח הדחף השקול מכלל המנועים הינו:

$$F_{tot} = F_1 + F_2 + F_3 + F_4 \quad (2.7)$$

את המומנטים הנוצרים סביב כל ציר כתוצאה מהמנועים ניתן לכתוב בצורה הבאה:

$$\begin{aligned} \tau_x &= l(F_2 - F_4) \\ \tau_y &= l(F_3 - F_1) \\ \tau_z &= \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{aligned} \quad (2.8)$$

כאשר l הינו המרחק של המנועים ממרכז הכותב של הכלי. את הכוחות והמומנטים ניתן להציג במטריצה בצורה הבאה:

$$\begin{bmatrix} F_{tot} \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & lb & 0 & -lb \\ -lb & 0 & lb & 0 \\ k & -k & k & -k \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.9)$$

את הקשר בין הכוחות למשתנים המתארים את תנועת הרחפן ניתן לפתח בשיטת ניוטון אוילר ובשיטת אוילר לגראנז', כאן יוצגו המשוואות המתקבלות על ידי שימוש בשיטת ניוטון אוילר.

המשוואות המתארות את התנועה הקווית של הרחפן במערכת אינרציאלית הינם:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \frac{1}{m} F_{tot} \begin{bmatrix} c(\phi) \cdot s(\theta) \cdot c(\psi) + s(\psi)s(\phi) \\ c(\phi) \cdot s(\theta) \cdot s(\psi) - s(\phi) \cdot c(\psi) \\ c(\theta) \cdot c(\phi) \end{bmatrix} \quad (2.10)$$

משוואות התנועה הזוויתית המתקבלות תחת הנחת זוויות קטנות הן :

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} I_x^{-1}[(I_y - I_z)\dot{\theta}\dot{\psi} + \tau_\phi] \\ I_y^{-1}[(I_z - I_x)\dot{\phi}\dot{\psi} + \tau_\theta] \\ I_z^{-1}[(I_x - I_y)\dot{\theta}\dot{\phi} + \tau_\psi] \end{bmatrix} \quad (2.11)$$

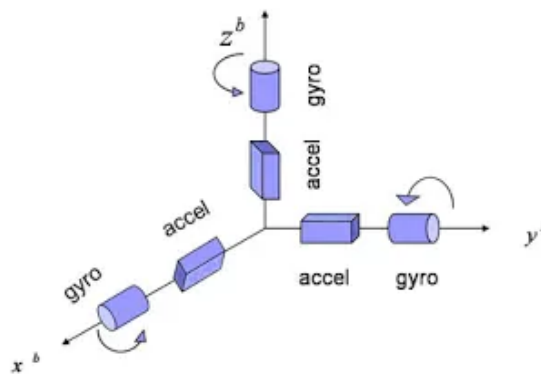
ניתן לשים לב כי מודל זה אינו לינארי ולכן תכנון מערכות בקרה הינו מורכב.

2.4. חיישנים

על מנת לאפשר שליטה מדויקת בכלי הטיס והוספת מערכות עזר למטיס, נדרשים מספר חיישנים המאפשרים מדידות של פרמטרים המייצגים את כלי הטיס ומאפשרים לקבל מידע על מיקומו במרחב ביחס לאובייקטים אחרים.

2.4.1. IMU

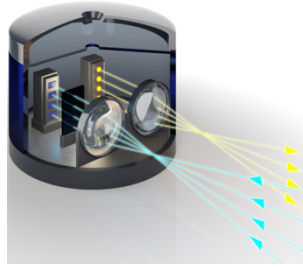
IMU הינו ראשי תיבות של Inertial measuring unit, אשר מוצג סכמטית באיור 3, הינו חיישן נפוץ בעולם הרובוטיקה, מכיוון שבעזרתו ניתן לקבל מדידות של המצב הזוויתי של הגוף. החיישן מורכב ממספר חיישנים- 3 חיישני גיירו אשר מודדים את המהירות הזוויתית של הרחפן בכל זמן נתון, ו-3 מדי תאוצה אשר מודדים את הכוחות הפועלים על הרחפן. בעזרת חיבור הקריאות משני סוגי חיישנים אלו ניתן לקבל מדידה יחסית מדויקת של המצב הזוויתי של הרחפן באוויר. קיימים גם חיישני IMU אשר בנוסף לגיירו ולמדי תאוצה, הם מורכבים גם ממגנטומטר אשר מודד את עוצמת השדה המגנטי מסביב לכלי הטיס, מידע המאפשר דיוק נוסף למדידות של המצב הזוויתי של כלי הטיס במקרים מסוימים. (Vectornav)



איור 3: תיאור של החיישנים והמדידות השונות מחיישן IMU

2.4.2. LiDAR

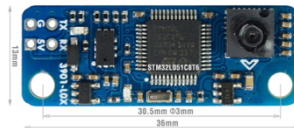
LiDAR (Light Detection and Ranging) הינו חיישן הפולט קרני לייזר כיווניים. לאחר פליטתם, מודד החיישן את הזמן הלוקח לקרני הלייזר לחזור וכך הוא מודד מרחק מאובייקטים. המרחק אותו יכול החיישן לאמוד תלוי בעוצמת קרן הלייזר אותה הוא פולט ובזמן ההמתנה של החיישן לקרן לחזור. יש מספר תצורות לחיישן מסוג זה, כאשר הנפוצה ביותר הינה חיישן הפולט קרן לייזר בכיוון אחד ומאפשרת חישה של המרחב בכיוון רצוי. תצורה נוספת הינה חיישן מסתובב אשר מאפשרת לקבל מדידות 360° מהגוף. (Matlab)



איור 4: אופן פעולה של חיישן LiDAR. בצהוב- פליטת גלים, בכחול- חזרתם לחיישן

2.4.3 Optical Flow

חיישן מאפשר זיהוי של תזוזות הרחפן בעזרת חיישן אופטי המזהה את שינוי המיקום של חלקיקים במשטח מעליו טס הרחפן, ביחס למיקומו. בעזרת מדידה זו, ומכיוון שהמדידה הינה יחסית לרחפן, ניתן לחשב את מהירות הרחפן בשני צירים- x, y . חיישן כזה, בתוספת חיישן LiDAR, הפונה כלפי מטה, מאפשר קבלת מידע מלא על גובה הרחפן ועל התזוזה הקווית שלו, ובכך ניתן לאפשר שהייה אוטונומית של הרחפן במיקום קבוע.



איור 5: חיישן Optical Flow בסיסי בעל חיבור UART

2.5.2 מנ

ישנם שני סוגים עיקריים של מנועים חשמליים- מנוע מברשות ומנוע ללא מברשות. בכלי טיב מסוג רב-להב משתמשים במנועים ללא מברשות מכיוון שהם מאפשרים ביצועים טובים, בעלי פינוי חום יעיל יותר ומאפשרים תחזוקה קלה יותר כתוצאה מכמות רכיבים נמוכה יותר. במנועים אלה קיימים מגנטים וסלילים. המגנטים בעלי קוטביות קבועה והם נמצאים בחלקו המסתובב של המנוע, הסלילים נמצאים בחלקו הנייח של המנוע ומחולקים לשלוש קבוצות בפאזה של 120° אחד מהשני. הפרש פאזה זה מאפשר רציפות בתנועת המנוע. מכך שקיימות קבוצות הסלילים, נדרש בקר שליטה נוסף אשר יודע לתזמן את העברת הזרם בסלילים. בקר זה נקרא *ESC – Electronic speed controller*, והוא מקבל פקודות מהירות וקובע את קצב העברת הזרם בין הסלילים השונים במנוע.

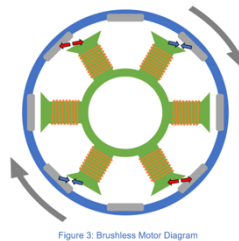


Figure 3: Brushless Motor Diagram

איור 6: סכימה של מנוע ללא מברשות. באדום- פאזה 1, בכחול- פאזה 2

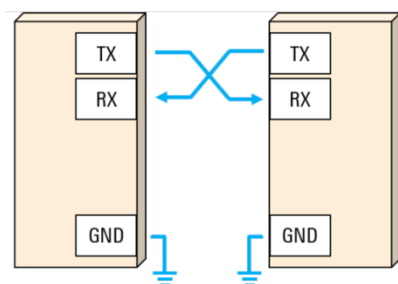
מהירות המנוע ברחפנים הינה פרופורציונלית למתח אשר נופל על המנוע. פרמטר זה נקרא kV , ועבור מתח מסוים, מנוע בעל kV גבוה יותר יסתובב מהר יותר.

2.6. חיבורי תקשורת

ישנם מספר פרוטוקולי תקשורת המאפשרים חיבור של מספר רכיבים על מנת שיוכלו להעביר מידע אחד עם השני. פרק זה יסקור את שני הפרוטוקולים הרלוונטיים לפרויקט זה.

2.6.1. UART

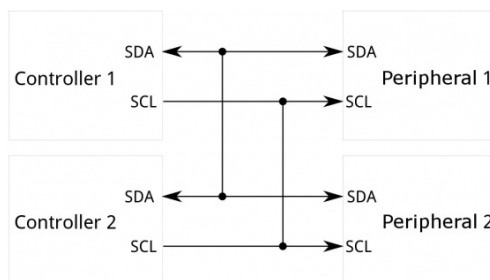
universal asynchronous receiver / transmitter או בקיצור UART, הינו פרוטוקול תקשורת המאפשר העברת מידע דו כיווני בין שני רכיבים. פרוטוקול תקשורת זה מורכב משני חיבורים Tx- Transmitter, Rx- Receiver. החיבור הראשון מיועד להעברת מידע החוצה מהרכיב והחיבור השני מיועד לקבלת מידע לרכיב. התקשורת יכולה להתבצע בשלוש קונפיגורציות שונות- מידע נשלח בכיוון אחד בלבד, מידע מועבר בשני הכיוונים אך בכל זמן נתון רק רכיב אחד מעביר מידע, מידע מועבר בשני הכיוונים ובאותו הזמן. (Rohde & Schwarz USA, Inc.)



איור 7: איור סכמטי של חיבור רכיבים על ידי חיבור מסוג UART

2.6.2. I2C

Inter-Integrated Circuit הינו פרוטוקול המבוסס על שני חיבורים SDA- Serial Data ו- SCL- Serial Clock. חיבור זה מאפשר תקשורת בין מספר רכיבים שונים עם בקר אחד. הרכיבים מחוברים לאותם חיבורי SDA ו- SCL בבקר. כל רכיב הינו בעל כתובת, ובעזרתה הבקר יודע לפנות אל החיישן הנדרש. (Sparkfun)



איור 8: איור סכמטי של חיבור מספר רכיבים לבקר אחד

2.7. סוללת $Li - Po$ וסוללת $Li - ion$

ישנם שני סוגים עיקריים של סוללות בהם נעשה שימוש עבור רחפנים. הסוג הראשון הינו סוללת $Li - Po$, והיא בעלת יכולת פריקה ($C rate$), גבוהה. מה שמאפשר תגובה מהירה יותר לדרישות המפעיל. סוללות מסוג זה משמשות בעיקר בתחום המרוצים ובהטסה מסוג $Freestyle$. הסוג השני הינו $Li - ion$, סוללות מסוג זה בעלות קצב פריקה נמוך יותר משמעותית בהשוואה לסוג הראשון. כתוצאה מכך הרחפן יגיב לאט יותר לדרישות המפעיל. בעקבות קצב הפריקה הנמוך, סוללות אלה מאפשרות זמן תעופה גבוה יותר, זהו היתרון

של סוללות אלה בהשוואה לסוללות $Li - Po$. בנוסף, ניתן לפרוק סוללת $Li - ion$ עד למתח נמוך יותר, ללא סכנה לפגיעה בסוללה, מאשר סוללת $Li - Po$ - 2.8V לעומת 3.5V (Robocraze).

2.8. תוכנות שליטה ברב-להב

כיום קיימות מספר רב של תוכנות אותן ניתן לצרוב על בקר הטיסה (FC) על מנת לאפשר שליטה ברחפן. התוכנות הנפוצות ביותר הן- *ArduPilot*, *PX4*, *BetaFlight*, *iNav*, כאשר כל תוכנה מתמקדת באופן טיסה אחר ובעלת חוזקות שונות. תוכנות אלה מכילות את הפילטרים, אמצעי השיערוך, קבלת מידע מחיישנים ומהשלט ומתן פקודות למנועים. התוכנות *ArduPilot* ו-*PX4* תומכות בחיבור של מחשב נוסף (*Companion PC*) ל-FC ובעזרתו ניתן לתכנת באופן עצמאי רובד שליטה נוסף ברחפן בהתאם לדרישות המתכנן. ישנם מספר בקרים בהם ניתן להשתמש כ- *Companion PC*, כאשר הנפוץ ביותר הינו *RaspberryPi*. על מיקרו בקר זה ניתן לתכנת אלגוריתמים בשפת תכנות *Python* המאפשרים שליחת פקודות ל-FC. קיימות מספר ספריות המקלות על תהליך שליחת הפקודות, בניחן, *DroneKit*, *MAVSDK*, *Pymavlink*, *mavROS*.

2.9. ספריית DroneKit

כלל תוכנות השליטה ברחפנים מתבססות על פרוטוקול תקשורת בשם *MAVlink*. פרוטוקול זה מעביר את הנתונים הנדרשים בין המפעיל לבין הרחפן. בין הנתונים קיים- זמן תעופה, מצב סוללה, מצב זוויתי ועוד. הספרייה *DroneKit* הינה ספרייה היודעת לקרוא את אופן הדיווחים, לנתחם ולהמיר לנתונים פשוטים הניתנים לשימוש. בנוסף, ספרייה זו מאפשרת מתן פקודות לרחפן ובכך לאפשר שליטה בעזרת מחשב נוסף השולח פקודות. על מנת לאפשר את מתן הפקודות נדרש שהרחפן יהיה במצב טיסה "*GUIDED*" מצב זה שומר על מיקום וגובה קבועים, ובמצב זה פקודות הטיסה מוגדרות להתקבל ממחשב אחר אשר מתכנן את מסלול הטיסה.

3. דרישות המערכת

על הרחפן בפרויקט זה לעמוד במספר דרישות. בחלק זה יוצגו הדרישות הראשוניות מהרחפן המפותח, עם התקדמות בפיתוח הפרויקט ייתכן שינוי בדרישות הרחפן.

3.1. גודל ומשקל הרחפן

מכיוון שהרחפן נדרש לטוס בשטחים סגורים בעלי אזורי תמרון קטנים, הרחפן נדרש להיות זעיר מספיק בשביל עבירות מקסימלית אך עדין מתן מענה על שאר דרישות המערכת. כמו כן, נדרש רחפן בעל משקל נמוך על מנת לאפשר נשיאה קלה, הרכבה ופירוק זריזים.

3.2. זמן תעופה

נדרש מהרחפן זמן תעופה מינימלי של חמש-עשרה דקות.

3.3. יכולות חישה

על מנת לאפשר פיתוח של רחפן אוטונומי נדרשים אמצעי חישה המאפשרים מדידה של מיקום הרחפן בסביבה בו הוא נמצא. נדרשת מדידה בכלל צירי הרחפן- X, Y, Z .

3.4. מערכות עזר

על מנת להתגבר על תופעות זרימה הנוצרות בסביבה צרה, ועל מנת לאפשר הטסה קלה יותר של המפעיל, נדרש פיתוח של מערכת בקרה המאפשרת שמירה על תנועה במרכז שטח הטיסה ביחס לצירים Y, Z . כמו כן המערכת תמנע מהרחפן להתקרב יותר על המידה לקירות.

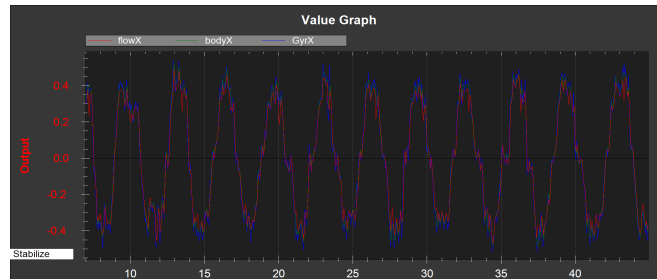
4. חלקי הפיתוח שבוצעו עד כה

הפיתוח מחולק לשני חלקים עיקריים- האחד, הרכבת הרחפן. חלק זה כולל הרכבה של החלקים הבסיסיים של הרחפן, צריבת התוכנה וחיבור החיישנים הנדרשים. השני, תכנון וביצוע של מערכת העזר למטיס. חלק זה יכול לתכנן מערכת בקרה לשמירה על מסלול רצוי. בשני חלקי הפיתוח יבוצעו ניסויים לבדיקת תקינות המערכות לפני התקדמות לשלב הבא.

עד כה התבצעו שלבים אשר ברובם שייכים לחלקו הראשון של הפרויקט- הרכבה. הרחפן נבנה על בסיס שלדה, FC ו- ESC של החברה *SpeedyBee*.

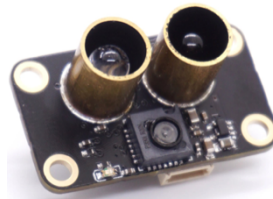
4.1. הרכבת הרחפן

עם תחילת הפרויקט נבנה רחפן המבוסס על FC מסוג *SpeedyBee F405 mini* אך לאחר מספר הטסות וזיהוי כי התקבלו מדידות לא אמינות מהרכיבים על הבקר, כמו גובה לא נכון או שגיאות ב*Gyro* שהובילו לקביעת מצב זוויתי לא נכון, הוחלט לבנות רחפן נוסף המבוסס על בקר טיסה אחר *SpeedyBee F405 V4*. לאחר בדיקות נוספות עם בקר טיסה זה התקבלו מדידות אמינות יותר. מכיוון שתוכנת ההטסה היחידה אשר תומכת בבקר טיסה זה הינה *ArduPilot* פרויקט זה יתבסס על תוכנה זו. כמו כן, חובר חיישן *Optical Flow* מסוג *Mateksys 3901 – L0X* אשר כולל גם חיישן *LiDAR* עד טווח של 2 מטר ובוצעו בדיקות לתקינות הטיסה. התקבלו תוצאות לא טובות אשר הובילו לרחפן לא יציב. על פי התיעוד של התוכנה *ArduPilot*, לאחר חיבור החיישן נדרש להשוות את ערכי המדידה של החיישן לערכי המדידה של הגייירו ושיתקבלו קריאות דומות.



איור 9: תוצאה רצויה מהשוואת Optical Flow ל- Gyro

בעזרת חיישן זה לא התקבלה התוצאה הרצויה של המדידות ולאחר מספר ניסיונות כיול התקבלה החלטה להחליף לחיישן Micoair MTF – 01, חיישן זה בעל חיישן זהה של Optical Flow אך בעל חיישן LiDAR עם טווח מקסימלי של 8 מטר.



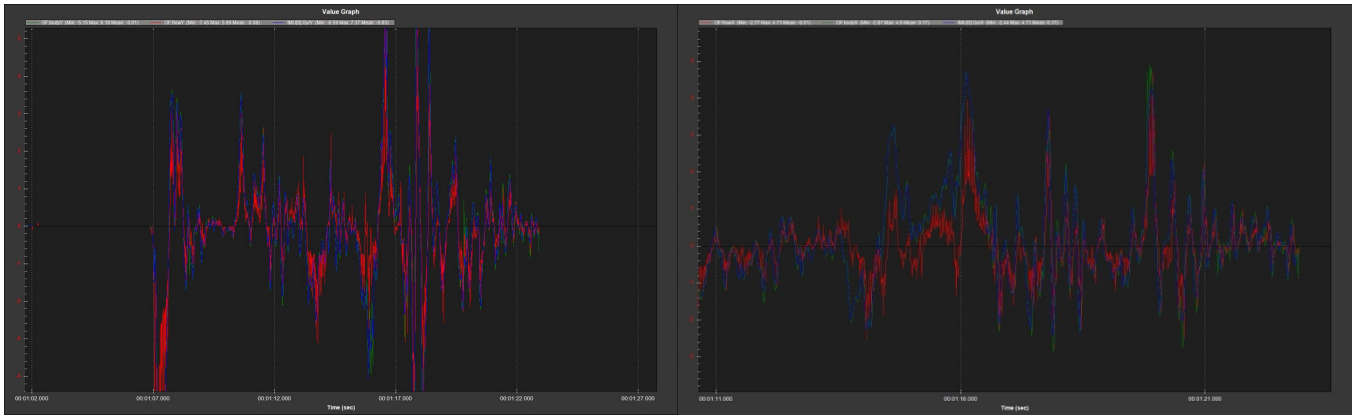
איור 10: חיישן Optical Flow Micoair MTF – 01

לאחר החלפת החיישן וביצוע ניסויים נוספים התקבלו הגרפים הבאים:



איור 11: השוואה בין ערכי ה- Gyro לערכי החיישן Optical Flow. מימין- ציר X, משמאל- ציר Y

מהגרף ניתן לראות כי ערכי Flow של החיישן (באדום) גדולים יותר בהשוואה לערכי הגייירו ולערכי ה Body אותו מודד החיישן (כחול וירוק בהתאמה). כמו כן, ניתן לראות כי ערכי הגייירו וערכי ה- Body דומים אחד לשני. לאחר זיהוי הבדל זה, הקיים בשני הצירים, התבצע שינוי לפרמטר המאפשר לכייל את היחס בין OF_{flow} ו- OF_{body} . פרמטר זה נקרא $FLOW_{FXSCALER}$, $FLOW_{FYSCALER}$ ישנם שני פרמטרים, פרמטר עבור כל ציר X, Y. לאחר כיול פרמטרים אלה וקביעת ערכם $FLOW_{FXSCALER} = 450$ ו- $FLOW_{FYSCALER} = 400$ התקבלה ההתנהגות הרצויה של הרחפן והיכולת לשמור על מיקום אך קיימות תנודות של הרחפן בציר Y, דבר המשפיע על יציבות הרחפן. נדרש תהליך כיולים של הגברי הבקרה בציר זה על מנת לאפשר התנהגות יציבה יותר של הרחפן.



איור 12: השוואה בין הערכים לאחר כיוול. מימין- ציר X, משמאל- ציר Y

רשימת רכיבים בהם יש שימוש בפרויקט לאחר ניסויים ובדיקת רכיבים שונים :

טבלה 1: רשימת רכיבים שנבחרו לאחר ניסויים

סוג רכיב	שם הרכיב	נתונים	חיבור חשמלי
Drone Frame	SpeedyBee Bee35	שלדת רחפן המתאימה לפרופלורים של 3.5 אינץ'	-
Flight Controller +ESC	SpeedyBee F405 V4 Stack	בקר זה מכיל בתוכו IMU, Barometer	בעל 6 חיבורי UART
Motors	SpeedyBee 2006 -1950KV	מנועים בעלי $r = 20 [mm]$ $h = 6 [mm]$ $KV = 1950$	מחובר ל- ESC, 3 חיבורים עבור שלושת הפאזות במנוע
LiDAR	Pololu VL53L1X	טווח מקסימלי של 8 מטר	מחוברים בעזרת I2C ל- RPI Zero
Optical Flow	Micoair MTF - 01	חיישן Optical Flow הכולל גם LiDAR בעל טווח מקסימלי של 8 מטר	מחובר ל- UART3 ב- FC
Companion PC	RaspberryPi Zero 2W	מחשב קל וקטן בעל יכולות עיבוד מספיקות לדרישות המערכת	מחובר ל UART2

בנוסף לתהליך חיבור החיישנים לבקר טיסה וכיוולם, חובר מחשב שליטה- RaspberryPi Zero 2W המאפשר שליחת פקודות על בסיס קוד Python. נכתב קוד בסיסי המתבסס בספריה DroneKit המאפשר תקשורת בין המחשבים וקבלת מידע, כגון מדידות החיישנים והמצב הזוויתי של הרחפן, למחשב השליטה מבקר הטיסה. בנוסף, התבצעו בדיקות של חיבור מספר חיישני LiDAR למחשב השליטה, קבלת מידע ומימוש פילטר אשר מעלה את רמת האמינות של המדידות. נציג את קטע הקוד אשר נכתב עד כה לביצוע התקשורת בין בקר הטיסה ל- RPI ובין ה- RPI לחיישני LiDAR.

הספריות הרלוונטיות לקוד זה הן :

```
1 import RPi.GPIO as GPIO
2 import VL53L1X as VL
3 from dronekit import connect, VehicleMode, LocationGlobalRelative
4 import time
```

איור 13: ספריות בהן יש שימוש בקוד פייתון

הספרייה *RPi.GPIO* הינה ספרייה אשר בעזרתה ניתן להגדיר מה השימוש בכל רגל של ה-*RPI*. האם מחובר חיישן אשר נדרש לקבל ממנו מידע או האם זה אקטואטור אשר נדרש לשלוח לו פקודות. השימוש בספרייה זו הינו בשביל להגדיר את הכתובות של חיישני ה-*LiDAR*. הספרייה *VL53L1X* הינה ספרייה אשר מאפשרת שליחת פקודות וקבלת מידע מחיישני ה-*LiDAR* בהם יש שימוש בפרויקט. הספרייה *dronekit* הינה הספרייה המאפשרת את התקשורת בין ה-*RPI* לבקר טיסה המריץ את התוכנה *ArduPilot*.

```
61 if __name__ == "__main__":
62     #Set pinout sensors:
63     XSHUT_front = 4
64     XSHUT_frontL = 5
65     XSHUT_frontR = 6
66     XSHUT_left1 = 7
67     XSHUT_left2 = 8
68     XSHUT_right1 = 9
69     XSHUT_right2 = 10
70     XSHUT_up = 11
71     XSHUT_back = 12
72     XSHUT_list = [XSHUT_front, XSHUT_frontL, XSHUT_frontR, XSHUT_left1, XSHUT_left2, XSHUT_right1, XSHUT_right2, XSHUT_up, XSHUT_back]
73     tof_front = 0
74     tof_frontL = 0
75     tof_frontR = 0
76     tof_left1 = 0
77     tof_left2 = 0
78     tof_right1 = 0
79     tof_right2 = 0
80     tof_up = 0
81     tof_back = 0
82     tof_list = [tof_front, tof_frontL, tof_frontR, tof_left1, tof_left2, tof_right1, tof_right2, tof_up, tof_back]
83
84     #Set address of the drone
85     address = 'udp:192.168.1.14550'
86     drone = UAV(address, XSHUT_list, tof_list)
```

איור 14: קטע קוד ראשי

זהו קטע הקוד הראשי, עקב מבנה הקוד בהמשך נדרש להגדיר מראש את חיבורי ה-*XSHUT* של כל חיישן *LiDAR*, זאת מכיוון שבעזרתם ניתן לשנות את הכתובת של כל חיישן. ונדרש לבצע הגדרה של משתנים אשר יאוכלסו במידע מהחיישנים. הסידור ברשימות הוא על מנת לאפשר עדכון אפקטיבי יותר. בשורה 85, הכתובת הנדרשת, אשר בעזרתה מתבצע החיבור בין הבקר טיסה למחשב, היא הכתובת אליה מחובר בקר הטיסה ל-*RPI*. המחלקה *UAV* הינה מחלקה המכילה בתוכה פרמטרים ופונקציות בסיסיים.

```
6 class UAV:
7     def __init__(self, address, XSHUT_list, tof_list):
8         self.address = address
9         self.vehicle = connect(address, baud=921600, wait_ready=True)
10        self.XSHUT_list = XSHUT_list
11        self.tof_list = tof_list
12        self.range = [0, 0, 0, 0, 0, 0, 0, 0]
13        self.range_p1 = [0, 0, 0, 0, 0, 0, 0, 0]
14        self.range_p2 = [0, 0, 0, 0, 0, 0, 0, 0]
15        #Enter the drone to PosHold mode
16        self.vehicle.mode = VehicleMode("POSHOLD")
17        # Weights for the LiDAR data filter
18        self.Kf1 = 0.3
19        self.Kf2 = 0.2
20        # Parameters to extract from the drone
21        self.pitch = 0
22        self.roll = 0
23        self.yaw = 0
24        self.velocity = 0
25
26        # Run LiDAR setup
27        self.setup_LiDAR()
```

איור 15: קטע קוד ראשי- מחלקת *UAV*

קטע קוד זה מבצע הגדרות כלליות הנדרשות לטובת הפעלת כלל הרכיבים. בשורה 9 נוצר החיבור בין ה-RPI לבקר טיסה. שורות 10-14 מבצעות השמה ואתחול של משתנים הרלוונטיים לחיישני ה-LiDAR, כמו כן, שורות 18 ו-19 מגדירות פרמטרים אשר ישמשו בפילטר המיושם על קריאות חיישני ה-LiDAR. שורות

```

29     def setup_LiDAR(self, range=2):
30         for i, XSHUT in enumerate(self.XSHUT_list):
31             GPIO.setup(self.XSHUT_list[i], GPIO.OUT)
32             GPIO.output(self.XSHUT_list[i], False)
33         for i, tof in enumerate(self.tof_list):
34             self.tof_list[i] = VL.VL53L1X(i2c_bus=1, i2c_address=0x29)
35             GPIO.output(self.XSHUT_list[i], True)
36             self.tof_list[i].open()
37             self.tof_list[i].change_address(0x30 + 2*i)
38             GPIO.output(self.XSHUT_list[i], False)
39             GPIO.setup(self.XSHUT_list[i], GPIO.IN)
40             self.tof_list[i].start_ranging(range)
41
42     #This function returns the distance from the LiDAR sensor in MM
43     def read_sensor_data(self):
44         for i, tof in enumerate(self.tof_list):
45             self.range[i] = self.tof_list[i].get_distance()
46         self.range = (1-self.Kf1- self.Kf2)*self.range + self.Kf1*self.range_p1 + self.Kf2*self.range_p2
47         self.range_p2 = self.range_p1
48         self.range_p1 = self.range
49
50
51     def extract_params(self):
52         while self.vehicle.mode.name != 'POSHOLD':
53             print('Waiting for vehicle to enter GUIDED mode...')
54             time.sleep(1)
55         self.roll = self.vehicle.attitude.roll
56         self.pitch = self.vehicle.attitude.pitch
57         self.yaw = self.vehicle.attitude.yaw
58         self.velocity = self.vehicle.velocity
59
60     def close_connection(self):
61         self.vehicle.close()
62

```

איור 16: פונקציות- מחלקת UAV

קטע הקוד המוצג מעלה מכיל את כלל פונקציות העזר אשר נבנו עד כה. הפונקציה המוגדרת בשורה 29 מבצעת השמה של חיישני ה-LiDAR לרשימה המוגדרת בשורה 11, כמו כן היא עוברת על כלל החיישנים ומבצעת שינוי כתובת. בשורה 43 מוגדרת פונקציה העוברת על כלל חיישני ה-LiDAR, מקבלת קריאות חדשות, מבצעת איחוד בין הקריאה הנוכחית לשתי קריאות קודמות על מנת לשפר את אמינות הקריאה (שורה 46). הפרמטרים המגדירים את המשקל הינם הפרמטרים המוגדרים בשורות 18-19. הפונקציה המוגדרת בשורה 51 מאפשרת קבלת מידע על המצב הזוויתי ועל המהירות של הרחפן מהבקר טיסה. בפרמטרים אלה יהיה שימוש בהמשך על מנת לממש את מערכת הבקרה אשר תשמש כמערכת עזר למטיס.

5. תכנית להמשך

החלק הראשון של הפרויקט, שלב בנית הרחפן, הסתיים לאחר תהליך הכיולים ומימוש היכולת שהיה במקום. המשך העבודה של הפרויקט יעסוק בכתיבת קוד המבוסס *Python* לטובת מימוש היכולת לנווט בעזרת חיישני ה-*LiDAR* המחוברים ל-*RPI*. כמו כן, יתבצע תהליך כיולים נוסף לקראת סוף הפרויקט על מנת לקבל הטסה נוחה יותר וללא תנודות. כעת הפרויקט יתרכז בפיתוח האלגוריתמים הנחוצים לטובת הסיוע למפעיל. הפיתוח יחולק למספר שלבים.

5.1. הפעלת חיישני *LiDAR*

שלב זה יתמקד בהפעלת חיישני ה-*LiDAR*, קביעת תנאים להסתמכות על קריאת החיישנים. במידה והקריאה גדולה מדי או קטנה מדי- לפלט. כמו כן, בניית אלגוריתם של *Observer Pattern*, אלגוריתם זה מאפשר קריאה דינמית של הערכים המתקבלים מהחיישנים ללא צורך בקריאה לפונקציות צדדיות.

5.2. קבלת מידע מבקר הטיסה ומתן פקודות

שלב זה יעסוק בהגדרת המידע הנדרש מבקר הטיסה לטובת מערכת הבקרה, כתיבת קוד *Python* המאפשר קבלת המידע ב-*RPI* מבקר הטיסה. קוד זה, בדומה לקוד המתוכנן בשלב 5.1, יתבסס על יכולת של הספרייה *DroneKit* לממש אלגוריתם דומה ל-*Observer Pattern*. שימוש באלגוריתמים אלה ייעל זמנים במהלך ריצת מערכת הבקרה ולא ידרוש שלב של עדכון פרמטרים כחלק ממערכת הבקרה.

5.3. תכנון מערכות בקרה וניווט

מערכת הבקרה תדרש לשמור על הרחפן במרכז המנהרה, כך הרחפן ישמור על מרחק מהדפנות וימנע את תופעות הטורבולנציה הגורמות להיפוכו. המערכת תיכנס לפעולה ברגע שהמפעיל מעביר את הרחפן למצב טיסה *FlowHold*, ניתן להכניס את הרחפן למצב זה לאחר המראה, במצב זה מערכות העזר הבסיסיות מתחילות לפעול ובכך גם מערכת הבקרה. כלל הפקודות של מערכת הבקרה ינתנו במערכת צירים צמודת גוף, זה יתבצע על ידי ציון מערכת הצירים הרצויה בקוד-*MAV_FRAME_BODY_OFFSET_NED*. מערכת צירים זו מגדירה כי הפקודות הניתנות הן במערכת הגוף, ביחס לכיוון ההתקדמות הנוכחי של הרחפן. אין פונקציה קיימת בספרייה *DroneKit* המאפשרת שליחת פקודות על מהירות סיבוב של המנוע. אך מכיוון שניתן לשלוח הודעות מהפורמט של *MAVLink* גם לא כחלק מפונקציה מוכנה, המשתמש יכול לבנות פונקציה עצמאית ועל ידי שימוש בפקודה *vehicle.message_factory.command_long_encode* ניתן לשלוח פקודות על מהירות סיבוב המנוע. ולכן אות היציאה של מערכת הבקרה יהיה מהירות סיבוב המנוע. אותות הכניסה יהיו המצב הזוויתי של הרחפן וקריאות חיישני ה-*LiDAR*.

מערכת הניווט בתצורתה הראשונית תהיה מבוססת על חיישני ה-*LiDAR*. בכל איטרציה של אלגוריתם הניווט המערכת תזהה את החיישן אשר זיהה את המרחק הכי גדול, כלומר הדרך הכי פנויה, הרחפן יפנה

להיות עם הפנים לכיוון זה ויתחיל תנועה במהירות מבוקרת וקבועה. בכל זמן הניווט האוטונומי מערכת הבקרה תפעל גם כן ותדאג לשמור את הרחפן במסלול המאפשר מניעה מהתקרבות לדפנות.

על מנת להתחשב בערך הסוללה יתבצע רישום של רמת הסוללה בעת התחלת הניווט האוטונומי ובכל איטרציה של המערכת תתבצע דגימה נוספת של ערך הסוללה ותתבצע השוואה ביחס לערך המינימלי של הסוללה אשר יקבע בהתאם לסוגה. לדוגמה- עבור סוללת $Li - Po \rightarrow V_{min} = 3.2 [V]$. כמו כן, כדי למזער את הסיכון של פריקה מלאה של הסוללה לפני יציאה מלאה מהמנהרה, ההשוואה תתבצע עבור ערך מינימלי של $V_{min} = 3.5 [V]$. ההשוואה תתבצע באופן הבא:

$$V_{start} - V_{curr} = \Delta V \quad (5.1)$$

$$if: V_{curr} - \Delta V == V_{min} \quad (5.2)$$

במידה והתנאי הרשום במשוואה 5.2 הרחפן יתחיל תנועה חזרה לנק' התקשורת האחרונה

5.4. מימוש מערכות הבקרה והניווט

לאחר שקיימים אלגוריתמים המאפשרים קבלת מידע מכלל החיישנים והרכיבים הנדרשים למערכות השונות, ולאחר תכנון של אופן הפעולה ייכתב אלגוריתם המפעיל את מערכות אלו. חלקים משלב זה יכולים להתקיים במקביל לשלב 5.3. עם התקדמות הפיתוח בשלב זה יתבצעו ניסויים לבדיקת תקינות המערכת ושיפור האלגוריתם.

6. ביבליוגרפיה

Understanding UART, Rohde& Schwarz, https://www.rohde-schwarz.com/us/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html#:~:text=UART%20stands%20for%20universal%20asynchronous,and%20receive%20in%20both%20directions

I2C, Sparkfun, <https://learn.sparkfun.com/tutorials/i2c/all>

Connect ESC & Motors, Ardupilot, <https://ardupilot.org/copter/docs/connect-escs-and-motors.html>

A Review of quadrotor UAV: Control and SLAM methodologies ranging from conventional to innovative approaches, Guray Sonugur,

<https://www.sciencedirect.com/science/article/abs/pii/S0921889022002317>

Inertial Measuring Unit, https://i1.wp.com/bestperformancegroup.com/wp-content/uploads/2012/05/Fig_5.jpg

Hermetic Packages for LiDAR Sensors, Schott <https://www.schott.com/en-th/products/hermetic-packages-for-lidar-sensors-p1000281/applications>

Optical Flow 3901-LoX, Mateksys, <https://www.mateksys.com/?portfolio=3901-lox>

How Brushless Motors work & How to test them, Emrik Joner, September 29, 2023, <https://www.tytorobotics.com/blogs/articles/how-brushless-motors-work>

Optical flow sensor testing and setup, Ardupilot, <https://ardupilot.org/copter/docs/common-optical-flow-sensor-setup.html>

MTF-01 Optical flow, RoboFusion, <https://robofusion.net/products/mtf-01-integrated-laser-range-finder-optical-flow-module-for-unmanned-aerial-vehicles-uavs>

Lithium-ion vs Lithium-polymer batter, Robocraze, <https://robocraze.com/blogs/post/lithium-ion-vs-lithium-polymer-battery>

Itroduction to LiDAR, MathWorks, <https://www.mathworks.com/help/lidar/ug/lidar-processing-overview.html>

What is inertial measuring unit, Vectornav, https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu?gad_source=1&gclid=CjoKCQjwxeyxBhC7ARIsAC7dS3_jltLGyyW0-08VLTvzWoPPf9VKRhuf3h71sUSJ_fGNqVMHY2IEKX4aAhADEALw_wcB