



מערכת בינה מלאכותית עבור רובוט לחיטוי מטוסים מקורונה

פרויקט מס' 2175-1-1-20

דו"ח סיכום



מגיש : אור בן-גָּל, 305445199.

מנחה : פרופ' יעקב פינגלרנט, אוניברסיטת ת"א.

תשפ"א



תוכן עניינים

תקציר	4
רקע – מגפת הקורונה	4
רובוט לחייטוי מטוסים מקורונה	4
מערכת AI עבור רובוט לחייטוי מטוסים מקורונה	4
דרישות מערכת ה-AI	4
הקדמה	5
מטרות הפרויקט	5
הרקע והמוטיבציה	5
הרובוט לחייטוי מטוסים מקורונה	5
הגישה לפתרון	6
רקע תאורטי	7
7 Simultaneous localization and mapping . 1	
בעיית ה-SLAM	7
דרכיים לשערוך פתרון בעיית ה-SLAM	8
Graph SLAM . 2	9
פתרון SLAM מבוסס אופטימיזציה	9
Graph SLAM	9
Real-Time Appearance-Based Mapping – RTAB-Map	10
Particle Filter . 3	10
פתרון SLAM מבוסס מסנן	10
Particle Filter	10
Rao-Blackwellized Particle Filter – Gmapping	11
Unscented Kalman Filter . 4	11
מסנן קלמן	11
Unscented Kalman ו- Extended Kalman Filter ,Kalman Filter	
Filter	11
שימוש ב- UKF לлокליזציה	12
Dynamic Window Approach . 5	13
IMPLEMENTATION	14
14 תיאור חומרה . 1	
דרישות המערכת	14
Nvidia Jetson Xavier NX	14
Intel RealSense D435i	15



16.....	2. תיאור תוכנה
16.....	Robot Operating System – ROS
16.....	מימוש ב-ROS
17.....	המימוש שלנו
18.....	סימולציה
18.....	הסביבה הייעודית
18.....	סביבת הסימולציה
20.....	רובוט
21.....	פתרונות הסימולציה
22.....	ביצוע מיפוי על סביבה אמיתית
24.....	ניתוח תוצאות
24.....	מיפוי תלת-מימדי (3D)
26.....	מיפוי דו-מימדי (2D)
29.....	סיכום מסקנות והצעות להמשך
29.....	סיכום
30.....	מסקנות
30.....	הצעות להמשך
31.....	מקורות
32.....	נספחים

רשימת איורים

4.....	איור 1 - דיאגרמת בלוקים
5.....	איור 2 - גלגל MECANUM
5.....	איור 3 - מנוע DROK Nema 17
6.....	איור 4 - חיפוי מלאה בין ה-LED בכל שטח הטבעת
6.....	איור 5 - בקר Arduino Mega 2560 rev3
6.....	איור 6 - דיאגרמת בלוקים (רובוט)
7.....	איור 7 - תצוגה מתמטית לבנייה-h-SLAM
8.....	איור 8 - Full SLAM (Offline)
8.....	איור 9 - Online SLAM
9.....	איור 10 - המACHINE מטריצית-L
9.....	איור 11 - דוגמה בשיטת GraphSLAM
10.....	איור 12 - Loop Closure
10.....	איור 13 - אלגוריתם של Particle Filter
11.....	איור 14 - עקרון פעולה של מסנן קלמן
12.....	איור 15 - עקרון השינוי במסנן קלמן המורחב (EKF)
12.....	איור 16 - השוואת בין UKF ו-EKF
13.....	איור 17 - פירוט מרכיבי DWA
13.....	איור 18 - DWA Trajectories
14.....	איור 19 - Nvidia Jetson Xavier NX
15.....	איור 20 - Intel RealSense D435i
15.....	איור 21 - Intel RealSense D435i Modules
16.....	איור 22 - מודל הפעולה של ROS עם שילוח צמתים כלליים
17.....	איור 23 - AI-qtf graph של מערכת
18.....	איור 24 - מודל תא נסעים של המטוס נוסעים הגדלן בעולם Airbus A380
18.....	איור 25 - תמונה של סביבת הסימולציה



19	איור 26 - סביבת הסימולציה מבט על
19	איור 27 - סביבת הסימולציה מנוקוד מבט פנימית
20	איור 28 - הרובוט בסימולציה
20	איור 29 - חומרת מערכת ה-AI בסימולציה
21	איור 30 - מיפוי דו מימדית של סביבת הסימולציה
21	איור 31 - מיפוי תלת מימדית של סביבת הסימולציה
22	איור 32 - המודל הפסיבי (כיסא) למיפוי סביבה אמיתית
22	איור 33 - מפה תלת מימדית עם מסלול המיפוי
23	איור 34 - תמונות מיפוי תלת מימדיות
23	איור 35 - תמונה 3D Occupancy Map
24	איור 36 - השוואה בין מיפוי תלת מימדי של סביבת הסימולציה לסביבת הסימולציה
25	איור 37 - מיפוי תלת מימדי מכמה זוויות והצגות עם סימוניים
26	איור 38 - השוואה בין מיפוי דו מימדי של סביבת הסימולציה לסביבת הסימולציה
27	איור 39 - השוואה בין מיפוי דו מימדי של סביבת הסימולציה ל-RTAB-Map + Gmapping
28	איור 40 - השוואה בין מיפוי דו מימדי של סביבה אמיתית ל-RTAB-Map + Gmapping
14	טבלה 1 - מפרט Nvidia Jetson Xavier NX
29	טבלה 2 - טבלת תוכרים

רשימת קיצורים

AI – Artificial intelligence

SLAM – Simultaneous localization and mapping

RBPF – Rao-Blackwellized Particle Filter

UKF – Unscented Kalman Filter

EKF – Extended Kalman Filter

SBC – Single Board Computer

DWA – Dynamic Window Approach

DoF – Degrees of Freedom

LiDAR – Light Detection And Ranging



תקציר

רקע – מגפת הקורונה

מגפת הקורונה היא מגפה עולמית מתמשכת של מחלת נגיף הקורונה. הנגיף, מזון-2-CoV-SARS, שעל פי הערכות עבר מבני חיים נגועים לאדם וגרם לחדוקתם של מאות מיליון בני אדם, מעריכים מעל 200 מיליון מאומתים, ולמאות של מעל 4 וחצי מיליון. התהਪצות הראשונה החלה בואהן שבסין, בדצמבר 2019, ובאמצע פברואר 2020 החלה להתרפש ברחבי העולם.

התפשטות המגפה והמחלה בעולם עיררה בהלה, מלואה במשמעות כלכלי משמעותי ובנפילה חריפה של הבורסות בעולם בחודש מרץ 2020. מדיניות העולם היגיבו למגפה באמצעות הנחיות לשמר על רוחוק חברתי, הטלת מגבלות על תנועה והתkekholiot, הטלת סגרים ווצרים, ביטול אירועים וסגירת כל המוסדות הלא חיוניים. כתוצאה לכך השتبש מהלך החיים התקין של מיליארדי אנשים שנפגעו בריאותית וככללית בכל מקום, ופגיעה בענף התעשייה בפרט.

רובוט לחיטוי מטושים מקורונה

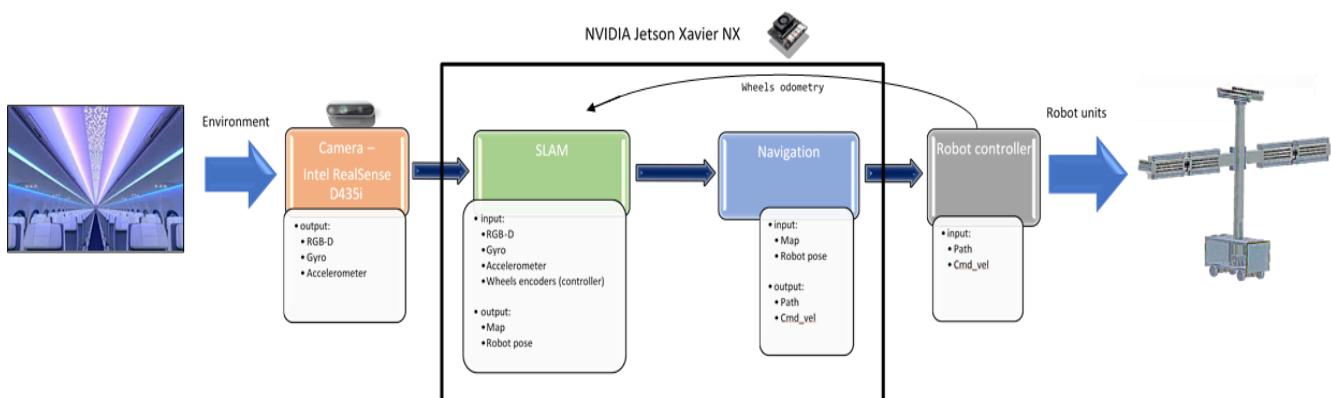
הפרויקט "מערכת בינה מלאכותית עבור רובוט לחיטוי מטושים מקורונה" הוא חלק מפרויקט גודל יותר "רובוט לחיטוי מטושים מקורונה". הרובוט מצויד בורות LED בתדר גבוה לחיטוי וירוסים ממושתחים כאשר הנורת מחוברת לזרועות שנפתחות ונסגורות לפחוץ.

הרובוט מתנשא על גבי גלגל MECANUM שמופעלים ע"י מנועים מסוד 17 Nema, וכן אוטם מנועים מפעילים גם את הזרועות ללא צורך בתמסורת, והבקר אשר שולט במנועי הרובוט הוא מסוג Arduino Mega 2560 rev3.

מערכת AI עבור רובוט לחיטוי מטושים מקורונה

מערכת הבינה המלאכותית מורכמת מ-*Nvidia Jetson Xavier NX* מסוג Single Board Computer-*m-D435i* (RGB-D) ומצלמת עמוק.

הרובוט דוגם את הסביבה בעזרת המצלמה אל המחשב, האותות עוברים לידי המחשב למיפוי הסביבה (תא הנוטעים במטוש) ושעירוך מיקום הרובוט באותה המפה שיהיה זמין למפעיל הרובוט. המפעיל ינות את הרובוט בחל המטוש תוך חיטוי המטוש מקורונה.



איור 1 - דיאגרמת בלוקים

דרישות מערכת ה-AI

על המערכת לאפשר ניוט של הרובוט בחלל המטוש מרוחק ע"י מפעיל או באופן אוטונומי, ובנוסך המערכת תמיפה בזמן אמת את חלל המטוש, ותספק אותן רלוונטיים למפעיל על מנת שיבין את תמונה המצב של הרובוט.



הקדמה

מטרות הפרויקט

- על המערכת לאפשר מיפוי מרוחק של חלל המטוס.
- על המערכת לספק כלים להבנת הסביבה למפעיל הרובוט, לרבות צילום וידאו מהרוביוט.
- על המערכת לשערך את מיקום הרובוט בmph שnochra.
- על המערכת לספק פלטפורמה לניווט הרובוט מרוחק.

הרקע והמושיבציה

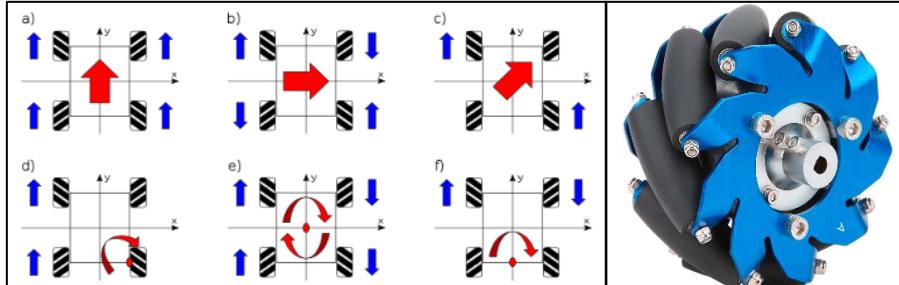
הנגייף עבר מאים לאדם כאשר נשא נושם החוצה נטו נשימוני ואירוסולים המרחפים באוויר, אשר מכילים את הוירוס. אדם אחר יכול לשומם פנים פנים את הנטו נשימוני והאיירוסולים. כמו כן הם יכולים לנחות על העיניים האף והפה, במיוחד במקורה של שיעול ועיטוש.

בתחילת המאה ה-20 ניסו למונע טיסה מהמוקד הראשון בסין ובמשך פסקו רוב הטיסות בין מדינות העולם ונסגרו הגבولات במועדים שונים. מדינות רבות הורו לבזק, לפחות תקופה מסוימת, את השבים אליהן ממדינות אחרות. זאת, במטרה למנוע הידבקות על ידי חולמים שאינם בעלי תסמיני קורונה.

סך הכנסות בענף התעופה העולמי ירדו ב- 69% ל- 189 מיליארד דולר בשנת 2020 (הפסדים נטו הסתכמו ב- 126.4 מיליארד דולר). נוצר צורך בענף התעופה להמשיך פעילות תקינה תוך שמירה על בריאות הציבור.

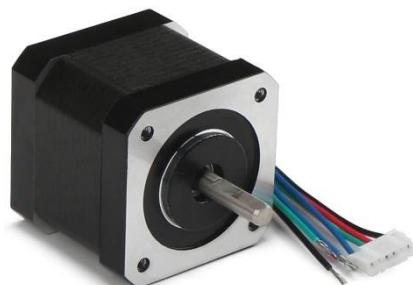
הרובוט לחיטוי מטוסים מקורונה

גלגלי MECHANUM – הגלגלים של הגלגל מסודרים בזווית של 45 מעלות לציר הגלגל, תצורה זו מאפשרת תנעה דינמית מהירה לכל כיוון, במיוחד לתנועה רוחבית שכן הגלגלים על גלגלי-MECANUM מעבירים במהירות וביעילות סיבוב גלגליים קדימה לתנועה הצדית. יש גם יתרון בעבודה הן להנעה והן להיגוי מכיוון שניתן להרכיב את הגלגלים הפונים לאותו כיוון.



איור 2 - גלגלי MECHANUM

DROK Nema 17 – מנוע סטפר בי פולארי מסווג שהמומנט המקסימלי שהוא יכול לעמוד בו הוא [Nm] 0.46 וכך ניתן להשתמש במנוע זה ללא שימוש בתמסורת. משקל המנוע קל יחסית – כ- 0.28 ק"ג ונitin לעובוד אותו במתוך של [V] 9. לרובוט חישה מנועים, ארבעהגלגלים ואחד לזרועות.

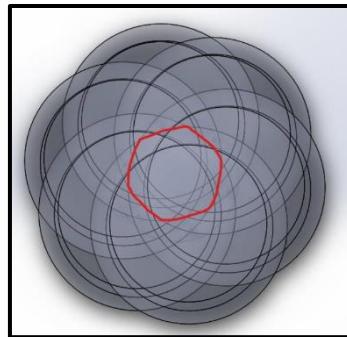


איור 3 - מנוע DROK Nema 17



נורות ה-LED

החייטוי יבוצע על ידי נורות לד בתדר גובה (אורך גל נМО) כ- 250nm . נורות ה-LED יורכבו על זרועות הרובוט כך שהרובוט מסירר במעבר זרועתו פורסota מעלה מכיסאות ומ�行操作. נורות ה-LED מסודרות במקבצים על מנת ליזור התאבכות של גלי האור לחיטוי מיטבי של המטחים והמכיסאות.

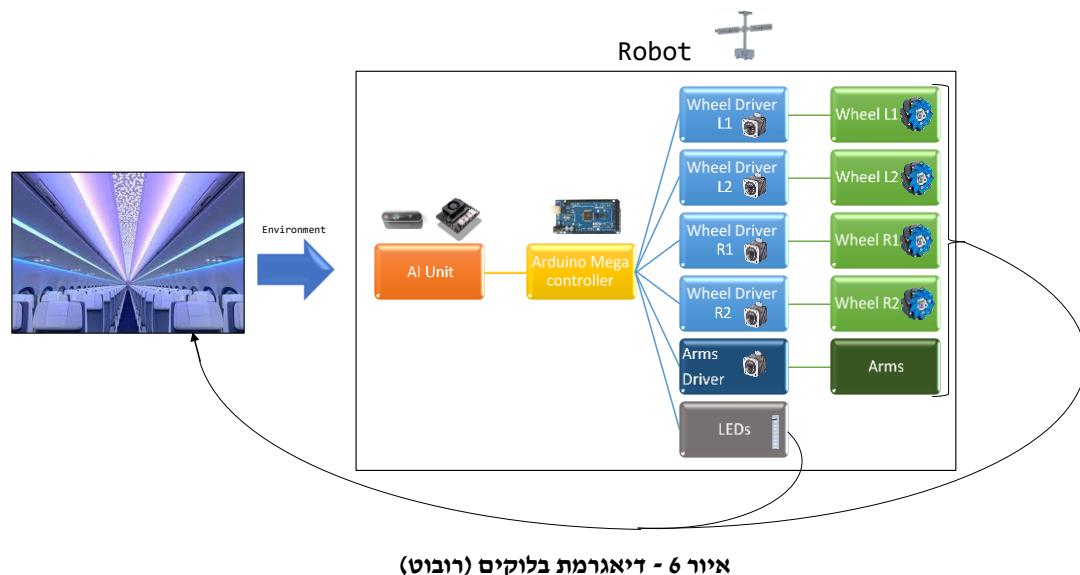


איור 4 - חפיפה מלאה בין ה-LED בכל שטח הטבעת

בקר 3 rev3 – תפקיד הבקר הוא שליטה בחלקי הרובוט המכניים לרבות מהירות הנסעה ופתיחה וסגירה של הזרועות. בנוסף הבקר מספק למערכת ה-AI-*Odometry* AI, על ידך קידוד תנועת הגלגלים.



איור 5 - בקר rev3



הגישה לפתרון

על מנת לעמוד המטרות הפROYיקט השתמשנו בפתרון בעיתת ה-SLAM (Simultaneous localization and mapping), שמאפשר לרובוט למצוא את מקומו במפה של כל המטוס ושרוך מיקום הרובוט במפה בזמן שהמפה נבנית. למפעיל אפשרויות לנווט ידני של הרובוט, הפעלת הרובוט על ידי לחצנים או>Nootot Chzi Autonomi bo a mafayil mesman leroboto. נקודת אליה הרובוט צריך להגיע והרובוט מנוט אל הנקודה, תוך הימנע מהתנגשות במכשולים.



רקע תאורי

הפרויקט עוסק באפליקציות מתחום הראייה ממוחשבת, למידת מכונה ועיבוד אותות. בחלק זהה נפרט על נושאי ה רק העיקריים.

Simultaneous localization and mapping .1

Simultaneous localization and mapping (SLAM) או בעברית מיפוי ואיICON בזמנית היא בעיה חישובית של הרכבה ועכון מפה של אזור לא ידוע, ובזמן שמירה על מיקום המפה (רוביוט במקורה שלו).

הנושא פופולרי ביותר בקרב קהילת הרוביוטיקה הנידית כבר יותר מ- 25 שנה. הצלחה של זה השדה קשור היטב לעובדה של "לפטור" את ענן ה-SLAM, כולם ביצעו לוקיזציה של רוביוט הוודות למפה של הסביבה הנבנית בהדרגה, יש יישומים רבים מהקירה מרחבית ועד נהיגה אוטונומית. הזורקרים האחרוניים שהו פנו לכלי רכב חכמים הרחיקו לכת את מאמצם המחקר בתרומה של יצרני הרכב.

בענית ה-SLAM נחשבת לאחד הדברים הכרחיים עבור רוביוטים אוטונומיים באמת, וכך הוא גם חיוני בהיבט של מכוונות בהניגה עצמית.

בענית ה-SLAM

כל המטריה היא להעריך במקביל את מצב הרוביוט והמפה הנבנית. ניתן להגדיר את מצב הרכב באופן שונה בהתאם לישום: מיקום וכיוון דו-מימדי (2D pose – x, y, z, roll, pitch, yaw), תנועת 6D (pose – x, y, z, roll, pitch, yaw), האצה וכיו'.

Given:

z_t – observation data (from sensor).

u_t – series of control inputs.

$$P(m_{t+1}, x_{t+1} | z_{1:t+1}, u_{1:t}) = ?$$

we want to estimate:

m_t – map of the environment.

x_t – agent's state (robot state).

sequential estimations (updates) of state and map posteriors:

$$P(x_t | z_{1:t}, u_{1:t}, m_t) = \sum_{m_{t-1}} P(z_t | x_t, m_t, u_{1:t}) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1} | m_t, z_{1:t-1}, u_{1:t})$$

$$P(m_t | z_{1:t}, u_{1:t}, x_t) = \sum_{x_t} \sum_{m_t} P(m_t | z_t, u_{1:t}, x_t, m_{t-1}) P(m_{t-1}, x_t | z_{1:t-1}, u_{1:t}, x_t)$$

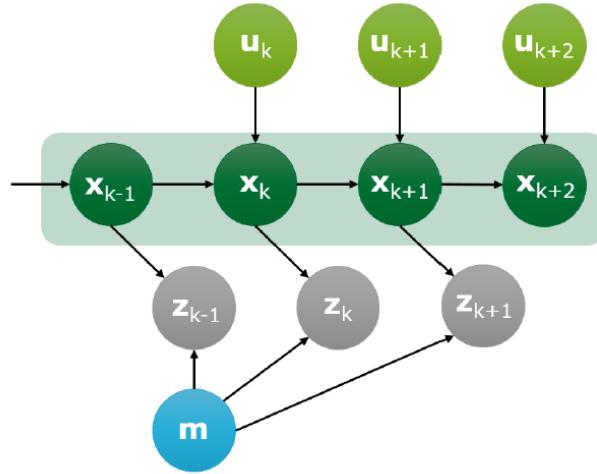
איור 7 - תצוגה מתמטית לבענית ה-SLAM

נסמן את שערוך המיקום והאוריינטציה של הרוביוט ב- x בזמן k , וב- m את שערוך מפת של הסביבה. על מנת לשערץ את המשתנים הללו משתמשים בנסיבות הבקשה, u , אשר מסייעות בשערוך התנועה בין הזמן $1 - k$ ו- k . כניסות הבקשה יגיעו ממקודדי הגלגלים (Wheel Odometry, WO) ומהיחסנים שכולים לתת מידע על מיקום כמו IMU. היחידה של הגישות לפתרונו SLAM הוא לחתוך בחשבו מדידות הנובעות מקריאת החישון שנסמן ב- z . אותן מדידות מסייעות לבנות ולשפר את המפה וביקיפין לשערץ את מיקום ותנועת הרוביוט.

באופן פורמלי ניתן לנתח את בענית ה-SLAM שתי דרכי. בדרך הראשונה, המטריה היא לאמוד את כל מסלול הרוביוט והמפה בהינתן כל כנויות הבקשה, u , וכל המדידות, z . הניסוח הזה ידוע כ-"Full SLAM", בבעיה זו מחשבים את ה-*joint posterior* על כל המיקומים והמפה בהתבסס על מכלול נתוני החישונים :



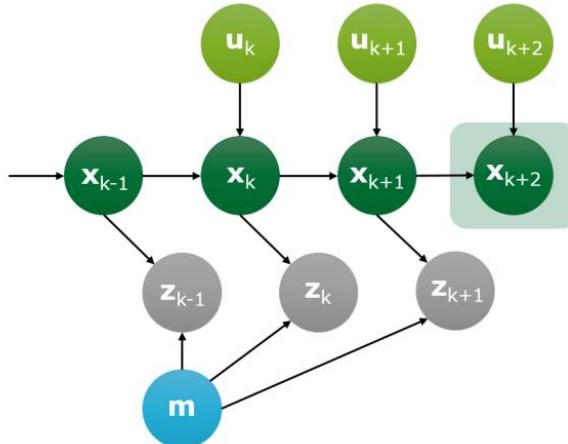
$$(1) \quad bel(x_{0:k}, m) = p(x_{0:k}, m | z_{0:k}, u_{0:k})$$



אייר 8 Full SLAM (Offline)

בדרכו השני הנקראת "Online SLAM", הרעיון הוא לשערך את ה-*pose* (מיקום ואוריינטציה) הנוכחי של הרובוט, על בסיס המידע האחרון שהגיע ממחישנים. ניתן להציג את האופי המסתבר של הבעה באמצעות חוק ביס:

$$(2) \quad bel(x_k, m) = p(x_k, m | z_{0:k}, u_{0:k}) \propto p(z_k | x_k, m) \int_{x_{k-1}} p(x_k | x_{k-1}, u_k) bel(x_{k-1}, m) dx_{k-1}$$



אייר 9 Online SLAM

זרכים לשערוך פתרון בעיית ה-SLAM

שעורך פתרון בעיית ה-SLAM מתחולק לשתי קטגוריות עיקריות: גישות מבוססות מסנן ושיטות מבוססות אופטימיזציה. הראשונה מתאימה לתהליכיים איטרטיביים ולכן מותאמת בעיקר לפתרון "Online SLAM" והשנייה מבוצעת עיבוד על מקבץ של נתונים ולפיכך מיושמת לרוב לפתרון "Full SLAM" או "Offline SLAM".



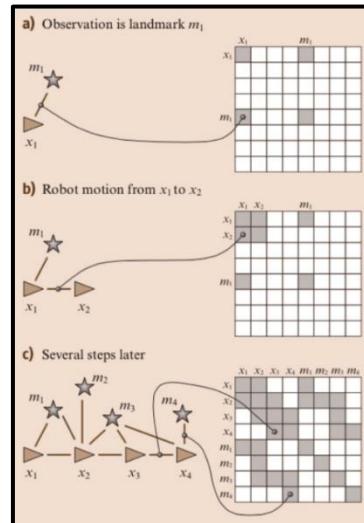
Graph SLAM .2

פתרון SLAM מבוסס אופטימיזציה

גישה פתרון SLAM מבוסס אופטימיזציה באופן כללי מורכבת שתי תת-מערכות, בדומה לפתרון מבוסס מסנן. תת-המערכת הראשונה מזוהה את האילוצים של בעיית ה-SLAM על בסיס המידע שמנגעו מהחישון ע"י התאמה בין הדגימות, z_k , והמפה. תת-המערכת השנייה מחשבת או מחדדת את ה-pose (וגם פוזיציות עבר) של הרובוט והמפה לאור האילוצים על מנת לקבל שילוב קוהרנטי.

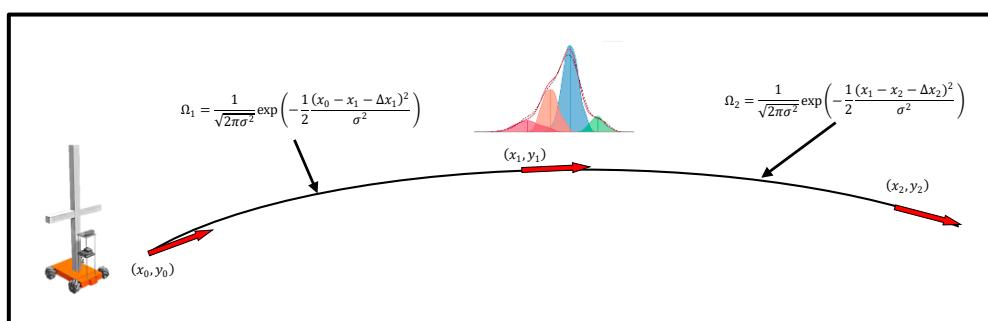
Graph SLAM

בהתבסס על ייצוג גרפי, מטריצה המתארת את היחסים בין סימוני דרך (landmarks) והפוזיציות של הרובוט ניתנת לבנייה בקלות יחסית ותשמש במסגרת שיטת האופטימיזציה.



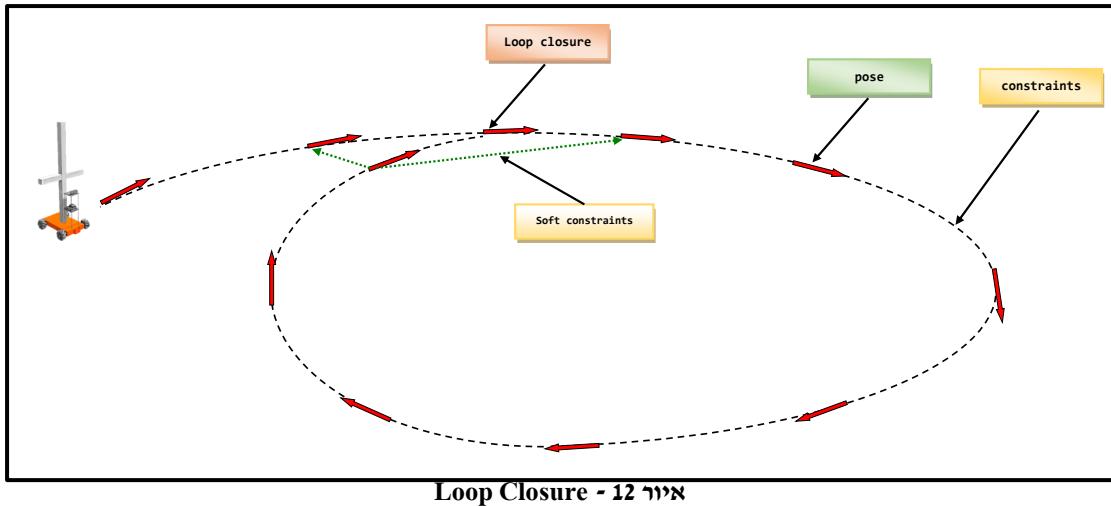
איור 10 - המחשה מטריצית ל-SLAM

בשיטה ה- Graph SLAM כשהרובוט בתנועה הוא דוגם מיקום ואוריינטציה (pose) לאורך הדרכ. בין כל שתי דגימות בעלות קשר (רציפות בדרך כלל) קיימות אי-ודאות מסוימות שבאה לידי ביטוי באילוצים (constraints). המטריה של השיטה הזאת היא בנייה גראף, כאשר הצמתים מייצגים את הפוזיציות של הרובוט, והקשות מייצגות את האילוצים. נרצה למצוא קונפיגורציה של צמתים כך שהשגיאה המյוצגת על ידי האילוצים מינימלית.



איור 11 - דוגימה בשיטת SLAM

כאשר הרובוט מגע לפוזיציה שכבר היה בה מתבצע סגירת לולאה, Loop Closure. ככלומר, הרובוט משווה דגימות לאורך פעולתו ובכך קיימת התאמה בין שתי דגימות האלגוריתם מבצע Loop Closure, על ידי זה שהוא מוסיף את האילוצים הרלוונטיים בין הצמתים הקשורים, ומתקן את המפה על ידי אופטימיזציה של צמתים להקטנת השגיאה.



אייר 12 - Loop Closure - 12

Real-Time Appearance-Based Mapping – RTAB-Map

RTAB-Map הוא SLAM RGB-D Graph based SLAM אשר מבצע גילוי Loop closure על מנת להחlijט את רמת הסבירות שתמונה מסוימת מראה. גלאי ה-Loop Closure משתמש בשיטת bag-of-words על מנת לחשוף את רמת הסבירות שתמונה מסוימת הגיעה ממיקום קודם או ממיקום חדש. כאשר השערת ה-Loop Closure מתקבלת על ידי האלגוריתם אילוצים חדשים נוספים לרף.

לאחר מכן הגרף עובר אופטימיזציה כדי להביא את השגיאות למינימום, ותהליך ניהול זיכרונות משמשת להגבלת מספר המיקומים המשמשים לגילוי Loop Closure ואופטימיזציית הגרף, כך שהאילוצים המתקבלים מהדגדימות גם בסביבה גדולה יותר יחסית יהיו משמעותיים.

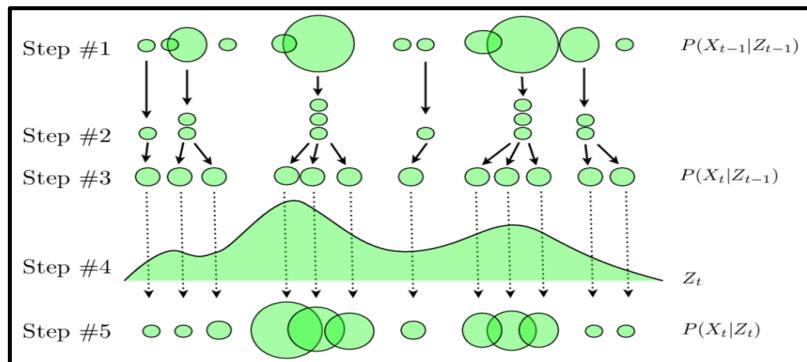
Particle Filter .3

פתרון SLAM מבוסס מסנון

שיטות המבוססות על מסננים נזירות מסינון בייסיאני ועובדות כתהליכי איטרטיביים דו-שלביים. בשלב הראשון מתבצע ניבוי של מצב הרובוט והמפה על ידי מודל שעורך וכניות בקרה, a_n . בשלב השני, המדידה הנוכחית שוגיאה מהחישון מושוואת נגד תיוקון למצב המשורך הקודם. המודל משווה בין המדידות למפה נקרא "Observation Model". שני השלבים האלה חוזרים ונשנים, ובאופן מctrber משלבים מידע מהחישון כדי לשערק את מצב הרובוט והמפה.

Particle Filter

אחד מהענפיהם העיקריים באlgorigthmi-h-SLAM הם פילטרים粒子滤波器 (PF). עיקרונו הפעולה של PF היא כלהלן: המצב נדגם עם חלקיקים לפי צפי ההסתברות שלו. לאחר מכן מתבצע חיזוי (likelihood) של כל חלקיק ועדכון מתבצע לפי הדגימה. בחלק של העדכון החלקרים מושקלים לפי ההסתברות (odds) שלהם על בסיס המדידה. החלקרים בעלי ההסתברות הגבוהה ביותר ייבחרו, ואשר החלקרים מוחלטים בחישובם. קיימות גרסאות רבות של PF, לכל אחת יתרונות וחסרונות לפי תכnuן וצורך, אך באופן כללי שיטות אלה דורשות כוח חישוב גדול יחסית.



אייר 13 - איטרטציה שלParticle Filter



Rao-Blackwellized Particle Filter – Gmapping

Rao-Blackwellized Particle Filter (RBPF) הוא אלגוריתם לשערוך פתרון בעיית SLAM אשר מבוסס עלParticle Filter כך שכל חלקיק מחזק מפה של הסביבה. הרעיון העיקרי הוא שערוך את הפוסטיריו (RBPF) Blackwellized Particle Filter מסלולים פוטנציאליים של $x_{1:t}|z_{1:t}, u_{0:t}$, ולחשתמש בפוסטיריו זהה לחישוב הפוסטיריו של המפה והמסלולים:

$$(3) \quad p(x_{1:t}, m|z_{1:t}, u_{0:t}) = p(m|x_{1:t}, z_{1:t})p(x_{1:t}|z_{1:t}, u_{0:t})$$

דבר שניתן לבצע ביעילות מכיוון שהפוסטיריו על המפה, בהינתן x ו- z , ניתן לחישוב באופן אנלטי.

אלגוריתם Gmapping שמיינשנו משתמש בלייזר וב-Wheel Odometry, והגרסה שמיינשנו מממשת טכניתה

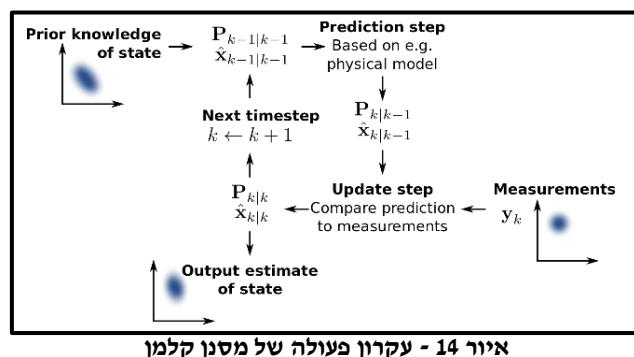
הנזרת M-RBPF Grid-Base FastSLAM MCL Particle Filter. גישת-h RBPF שකראת FsatSLAM כדי לשערך את הפוזיציה (локליזציה) ובנוספ' כל חלקיק מתחזק מפה משלו לטובת באופן איטרטיבי, ככלומר, RBPB, כדי לשערך את הPOSTIORIOOR הווה לחישוב שערוך מפה.

Unscented Kalman Filter .4

מסנן קלמן

מסנן קלמן הוא אלגוריתם, המכבל סדרה של מדידות מושעות של ערכאים שונים המיצגים מצב של מערכת בזמן מסוים, ומיציר הערכה, בדרך כלל מדויקת יותר, של אותן הערכאים על סמך תוצאת המסנן בעבר. המסנן עובד בצורה רקורסיבית על קלט מושע ומיציר הערכה סטטיסטית של מצב המערכת.

אלגוריתם מכיל מערכת משוואות והוא עובד בשני שלבים: ראשית, האלגוריתם מיציר הערכה של מצב המערכת (כלומר, ערכי המדדים שלהם) ושל ערכי אי הווודאות שלהם. בשלב השני, נקלוטות התוצאות של המדידות הבאות (בדרכ' כולל שגיאות מדידה מסוימות). תוצאות אלו מתקבלות כל אחת משקל מלאה והוא מעדכן באמצעותם את מצב המערכת. משקל כל תוצאה תלוי ברמת האמיןות אותה מייחס האלגוריתם למדידה. המשקלים יכולים להשתנות במשך פעולה המסנן בהתאם לתוצאות המתקבלות בכל מדידה. כך למשל, במידה שמתיקבל הפרש גבוה בין ההערכה לבין המדידה, המסנן עשוי להפחית את המשקל שהוא נותן להערכת הקודמות שלו ולהגדיר את המשקל הנitin לתוצאה, ולהפוך.



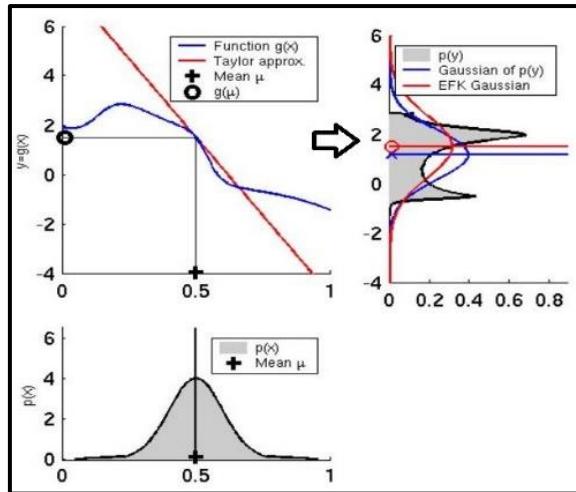
איור 14 - עקרון פעולה של מסנן קלמן

Unscented Kalman Filter-ו Extended Kalman Filter ,Kalman Filter

המגבילות במסנן קלמן הן שהוא רק עם התפליגויות גאוסיות ופונקציות לינאריות, דבר שלא ממש מתקיים עבור נתוני מציאות כמו מיפוי ולוקליזציה.

מסנן קלמן מוחרב (EKF) הפך לטכנית סטנדרטית המשמשת במספר יישומי שערוך לא לינארי ולמידת מכונה. אלה כוללים הערכת המצב של מערכת דינמית לא לינארית, אומדן פרמטרים ליזחי מושכת לא לינארית (למשל, למידת המשקלות של רשת עצבי), ושערוך כפול (למשל, אלגוריתם EM) שבו שני המצבים והפרמטרים מוערכים בו זמנית.

מסנן קלמן המוחרב מתמודד עם בעיות הלינאריות של מסנן קלמן על ידי שערוך הפונקציה (פונקציית החיזוי, המיצגת את היחסים בין ה-states (states- טו טיליר מסדר 1, ככלומר מבצעים שימוש בפונקציות לינאריות בפיתוח טיליר של הפונקציה הלא-LINARITY באזורי התווחת, ואז משערך את הערכים הרצויים).



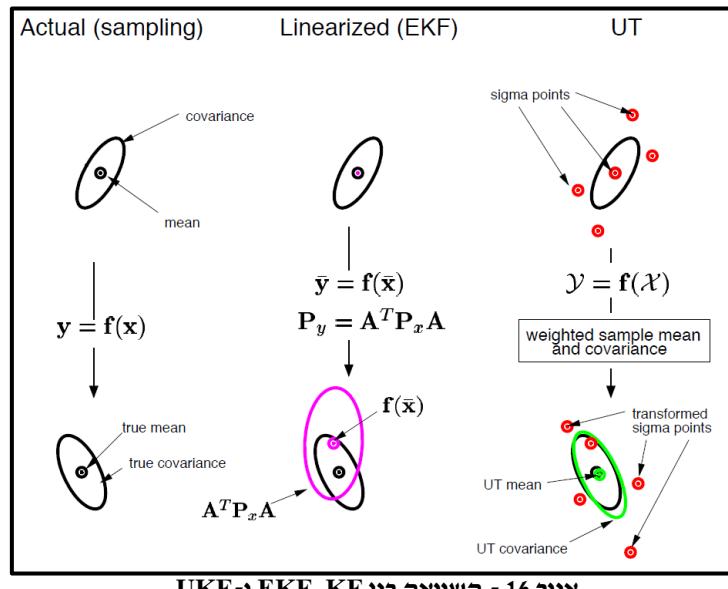
איור 15 - עקרון השינוי במסנן קלמן המורחב (EKF)

שימוש ב-UKF לлокציה

כפי שניתן להבין גם במסנן קלמן המורחב קיימים חסרונות מסוימים בגללlianאריזציה של הפונקציה לא ליניארית סיבוב התוחלת בלבד, ובעקבות כך הביצועים של EKF לא טובים יחסית. לעומת זאת במסנן קלמן "לא ריח" (הסביר על השם ב-[19]), נתייחס אליו UKF. UKF משתמש Unscented Transform, טרנספורמציה שמשמשת לשערץ תוצאות של החלת פונקציה לא-ליניארית נתונה על התפלגות שמאופיינת בסטטיסטיקה מסדר סופי. ב-UKF לעומת זאת השערץ מבצע עבור מספר נקודות שנקראות Sigma Points ומייצגות את כל ההתפלגות.

: (UT) Unscented Transform

- .1. חישוב סט של נקודות סיגמא (Sigma Points).
- .2. משקל נקודות הסיגמא.
- .3. החלת פונקציה לא-ליניארית על נקודות הסיגמא.
- .4. חישוב גאוסיאן מהנקודות לאחר משקל וטרנספורמציה.
- .5. חישוב תוחלת ושונות של הגאוסיאן החדש.



איור 16 - השוואה בין UKF, EKF, KF



Dynamic Window Approach .5

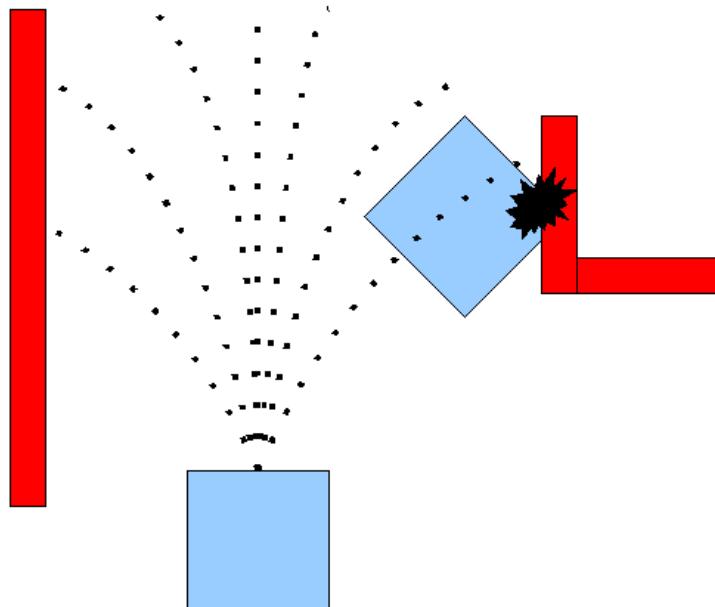
אוף ניוט הרובוט כאשר הוא מתבצע באוף חci אוטונומי מותבצע בעזרת גישת החלון הדינמי. גישת החלון הדינמי (DWA) היא אסטרטגיית מניעת התנגשויות (collision avoidance) מקוונת עבור רובוטים ניידים שפותחה על ידי דיטר פוקס, ולפרם ברגארד וסבטיאן ת'ירון בשנת 1997. בוגוד לשיטות הימנענות אחרות, גישת החלון הדינמי נגזרת ישירות מזהדינמיקה של הרובוט, והיא תוכנה במיוחד להטמודד עם האילוצים המוטלים על ידי מהירות מוגבלת ותאונות של הרובוט.

הוא מורכב משני מרכיבים עיקריים, הראשון מייצר מרחב חיפוש תקף, והשני בחירת פתרון אופטימלי במרחב החיפוש.

1. **Search space:** The search space of the possible velocities is reduced in three steps:
 - (a) **Circular trajectories:** The dynamic window approach considers only circular trajectories (curvatures) uniquely determined by pairs (v, ω) of translational and rotational velocities. This results in a two-dimensional velocity search space.
 - (b) **Admissible velocities:** The restriction to admissible velocities ensures that only safe trajectories are considered. A pair (v, ω) is considered admissible, if the robot is able to stop before it reaches the closest obstacle on the corresponding curvature.
 - (c) **Dynamic window:** The dynamic window restricts the admissible velocities to those that can be reached within a short time interval given the limited accelerations of the robot.
2. **Optimization:** The objective function

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega))$$
 is maximized. With respect to the current position and orientation of the robot this function trades off the following aspects:
 - (a) **Target heading:** *heading* is a measure of progress towards the goal location. It is maximal if the robot moves directly towards the target.
 - (b) **Clearance:** *dist* is the distance to the closest obstacle on the trajectory. The smaller the distance to an obstacle the higher is the robot's desire to move around it.
 - (c) **Velocity:** *vel* is the forward velocity of the robot and supports fast movements.
 The function σ smoothes the weighted sum of the three components and results in more side-clearance from obstacles.

איור 17 - פירוט מרכיבי DWA



DWA Trajectories - 18



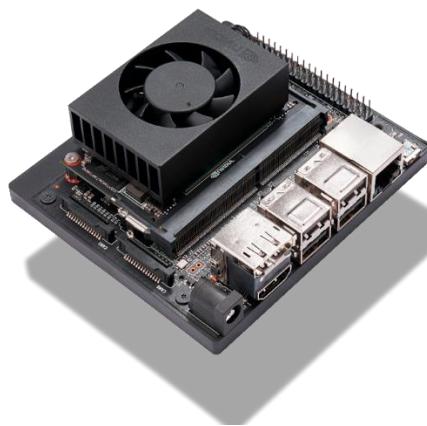
מימוש

1. תיאור חומרה

דרישות המערכת

חלק מדרישות הפרויקט, מערכת AI-TCR יכולה לעבוד גובה FPS אחד לפחות למספר GPU מסוים לעובדה Real-Time. בנוסף המערכת תבצע שני אלגוריתמי SLAM (Gmapping, RTAB-Map), לוקליזציה בערתת UKF ותעב אתות עומק, Gyro, RGB, Accelerometer ו-Wheel Odometry. בנוסף המערכת תבנה ותתוחזק מפה ותשלח פקודות ניווט לפי הצורך.

Nvidia Jetson Xavier NX



איור 19 - Nvidia Jetson Xavier NX

Developer Kit Technical Specifications	
GPU	NVIDIA Volta™ architecture with 384 NVIDIA® CUDA® cores and 48 Tensor cores
CPU	6-core NVIDIA Carmel ARMv8.2 64-bit CPU 6 MB L2 + 4 MB L3
DLA	2x NVDLA Engines
Vision Accelerator	7-Way VLIW Vision Processor
Memory	8 GB 128-bit LPDDR4x 59.7GB/s
Storage	microSD (Card not included)
Video Encode	2x 4K60 4x 4K30 10x 1080p60 22x 1080p30 (H.265) 2x 4K60 4x 4K30 10x 1080p60 20x 1080p30 (H.264)
Video Decode	2x 8K30 6x 4K60 12x 4K30 22x 1080p60 44x 1080p30 (H.265) 2x 4K60 6x 4K30 10x 1080p60 22x 1080p30 (H.264)
Camera	2x MIPI CSI-2 D-PHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E (WiFi/BT included), M.2 Key M (NVMe)
Display	HDMI and DP
USB	4x USB 3.1, USB 2.0 Micro-B
Others	GPIOs, I2C, I2S, SPI, UART
Mechanical	103 mm x 90.5 mm x 34 mm

טבלה 1 - מפרט Nvidia Jetson Xavier NX

Single Board Nvidia Jetson NX הוא SBC (Single Board Computer) מהמתקדמים בשוק הקמענאי.

לאור דרישות המערכת אנחנו צריכים SBC בעל כוח חזוף מספק על מנת שנוכל להריץ את שני אלגוריתמי SLAM, תוך העיבוד רב של מידע והעברתו בין התהליכים השונים, עם זיכרון של 8GB ומטמוניים 2: 4MB RAM ו- 6MB L2 ו- 4MB L3: יותר מפי 2 מה-SBC המקבילים.

בנוסף ל-Jetson Xavier- יכולת עיבוד גרפית גבוהה שתסייע בעיבוד הוויזואלי המקבילי, והוא נבנה באופן כזה שתומך במספר רב של אפליקציות AI.



Intel RealSense D435i

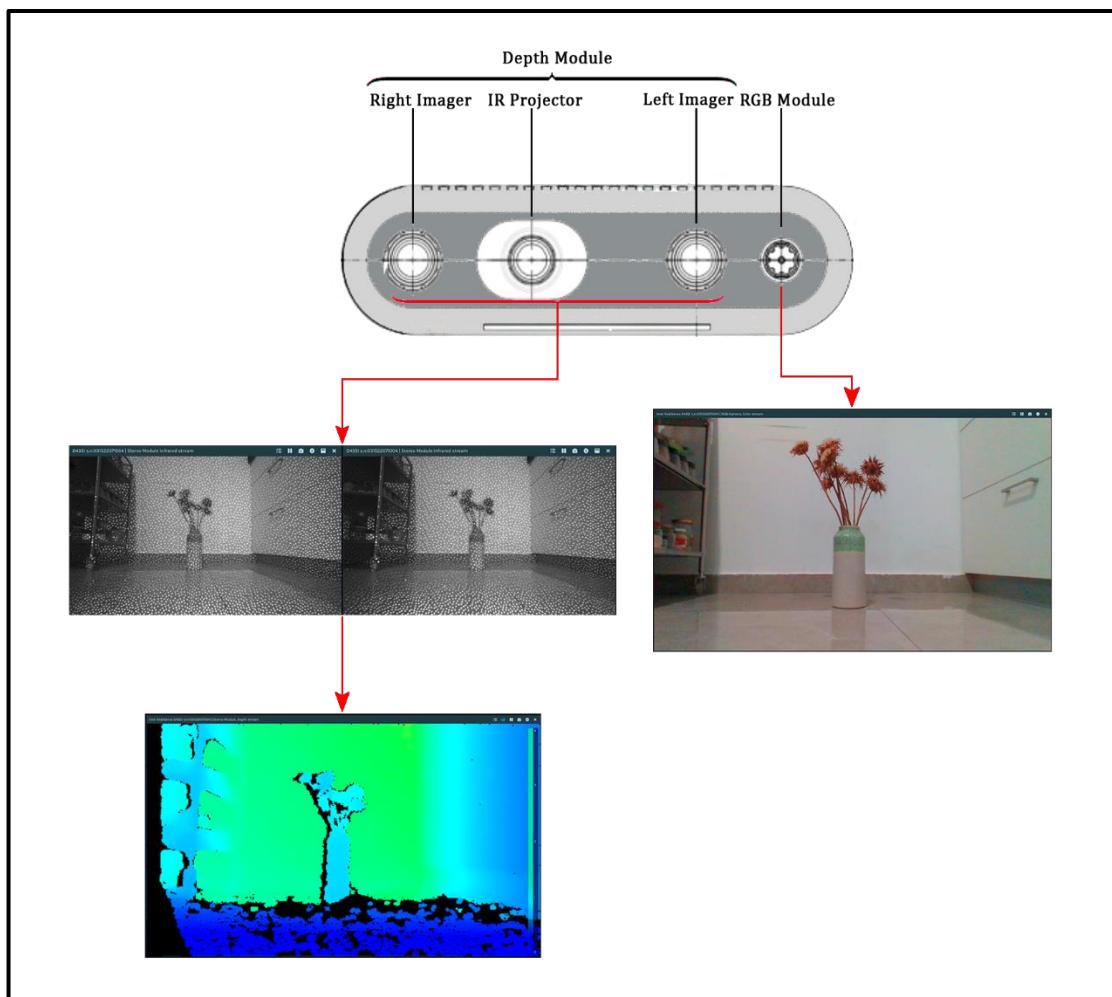


איור 20 Intel RealSense D435i - 20

על מנת שנוכל למפות באופן תלת-מימדי עבור הרובוט ובאופן דו-מימדי עבור הרובוט (לниוטט), השתמשנו במכצלמה D435i. זהה מכצלמת עומק אשר מספק תכונות עמוק וצבע, ובנוסף במכצלמה יש IMU מובנה עם 6 Degrees of freedom (6-DoF).

זהה מכצלמה אופטימלית למיפוי RGB-D RTAB-Map עם RTAB-Map, ובנוסף ניתן להמיר את תמונה העומק שלה לסדריות ליזר לטובת מייפוי Gmapping.

המכצלמה מספק מדידות של IMU (Gyro, Accelerometer) לשעורך לוקלייזציה.



איור 21 Intel RealSense D435i Modules - 21



2. תיאור תוכנה

Robot Operating System – ROS

ROS היא מערכת Open-Source מוכוונת רובוטיקה אשר מתפקדת כ-Meta-Operating System, ומעניקה שירותים הדומים לשירותי מערכת הפעלה הכלולים הפשוטות חומרה, בקרה של התקנים Low-level, מימוש פונקציונליות בשימוש רב, תקשורת בין תהליכיים, וניהול חבילות. ROS לא מחליף מערכת הפעלה, אלאעובד יד ביד עם מערכת הפעלה.

ב-ROS ניתן לכתוב ולהריץ קוד מבוזר במחשבים שונים שכותוב בשפות שונות למשל C++, פיתון ועוד.

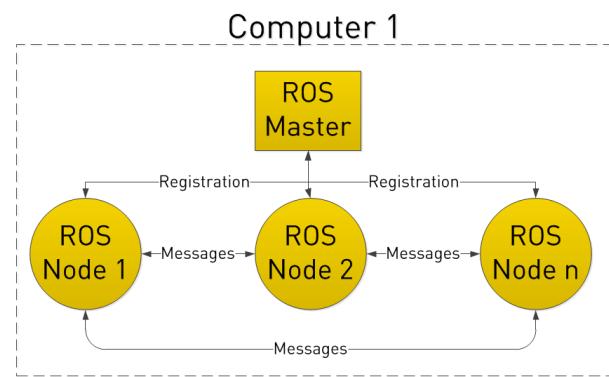
אחד הסיבות לכך שפיתוח תוכנה לרובוטים לרוב מטגר יותר מפיקוחים אחרים היא שהבדיקות עשוות להיות ארכוכות עם נטייה לשגיאות. לא תמיד רובוטים פיזיים זמינים לעבודה, ואחרם הם כן, התהיליך לפעמים איטי ומורכב.

ב-ROS ניתן להפריד בין השליטה הישירה ברמת ה-Low-level בחומרה, ועיבוד וקבלת החלטות ברמת ה-High-level לתוכניות ותהליכיים נפרדות ותהליכים נפרדים. בזכות ההפרדה הזאת, ניתן להחליף זמינות את מערך ה-Low-level, הכול חומרה ותוכניות בקרה למשל, בסימולטור על מנת לבדוק את ההתנהגות של מערך ה-High-level הרובוט.

משמעות העברת ההודעות של ROS הופך לסטנדרט בפועל עבור יכולת פעולה של תוכנת רובוט, כמו רום ROS מתממשק להומרה העדכנית והן ליישומים של אלגוריתמים מתקדמים אשר לעתים קרובות זמינים.

אחד המטרות של ROS היא תמכה בשימוש קוד קיים (שימוש חוזר) במקורה ופיתוח על ידי חלוקה לחבילות, כך שייהיה ניתן להשתמש בהן באופן מובנה.uproject זהה שימוש בשני אלגוריתמי SLAM מורכבים באופן משולב היה ניתן למימוש הודות לתכונה הזאת.

בנוסף ל-ROS כלים וספריות המסייעים לפיתוח הרובוט, ביניהם כלים לוויוואלייזציה, בקרה ותקשורת.



איור 22 - מודל הפעלה של ROS עם שלושה צמתים בלבד

מימוש ב-ROS

כאשר מפעילים פרויקט ב-ROS זה מתחילה בפיתוח ROS Master. ROS Master הוא התוכנה שמספקת שירותים לשאר הצמתים (Nodes) של ROS. כל הפיקוד והשליטה ב-ROS נעשית על ידי התהיליך (Process) הזה. כאשר שני צמתים מתחברים על ידי Topic או Service או יידי תהליך ה-ROS Master עוכב אחר התקשרות הזאת, מספק שירותים מתאימים ומסייע לצמתים לאתר אחד את השני.

את התקשרות בין צמתים ניתן לחלק לשני אופנים עיקריים :

Topics – שם תקשורת מסווג ו- Publishers Subscribers Topic. למשל ב-SLAM המצלמה מפרסמת (Publish) את הפרמטרים שלה ל-Topic ואלגוריתמי ה-SLAM משתמשים בפרמטרים אלה, מאזינים או מננים (Subscribe) לאותו Topic.

Services – הם סוג תקשורת חד צדדי, באופן של בקשה ותשובות. למשל שירותים (Service) ניתן בדרך כלל על ידי צמתים זהה תפקדים, צומתי שירות. כאשר צומת מסוימת צריכה שירות מצומת השירות היא שולחת בקשה, וכתשובה מקבל שירות מהצומת. קיימים מספר שמות לייחסים בין הצמתים : Server-Client, Talker-Listener .
דוגמה לרלוונטיות לשירות שמסופק ב-ROS היא Nav msgs (הודעות ניווט), כאשר צמת הלוקה מבקשת ניווט לנקודות ציון מסוימות ((x_i, y_i)) וצומת השרות, במקרה הזה בדרך כלל Move_base, מספקת את השירות על ידי צמתים נוספים.

אחד ההבדלים העיקריים בין Topics ל-Services הוא אופן התזמון, Topics משתמשים לפונקציונליות רציפה כמו תזרירם של פרימיטים, או מיקום, מידע שיפורסם ומתקדם אופן רציף. ולעומתם Services נותנים מענה אופן נקודתי, כמו קבלן עבודה וחוזרים להמתין לבקשת הבאה.



הIMPLEMENTATION שלנו

בפרויקט זהה מימושו ב-ROS שני אלגוריתמי SLAM באופן הבא:

- מימוש באופן RGB-D Odometry, האלגוריתם פוזיציון מהסביבה, בנוסף לכך הוא מקבל מ-Topics Wheel Odometry של המצלמה. האלגוריתם כותב ל-Topic /map, שהוא מפה 2D, ול-Topics נוספים כמו /MapCloud

+ 2D LiDAR (Light Detection And Ranging), Gmapping – מימוש באופן היחיד שנותנו למשמש אותו, Odometry, והמנת שוכן לספק פרימיטים של סריקת ליזר, השתמשתי במירר תמינות עמוק לסריקות ליזר Topics (depthimage_to_laserscan) כותב ל-Topic /map ו-Topics נוספים.

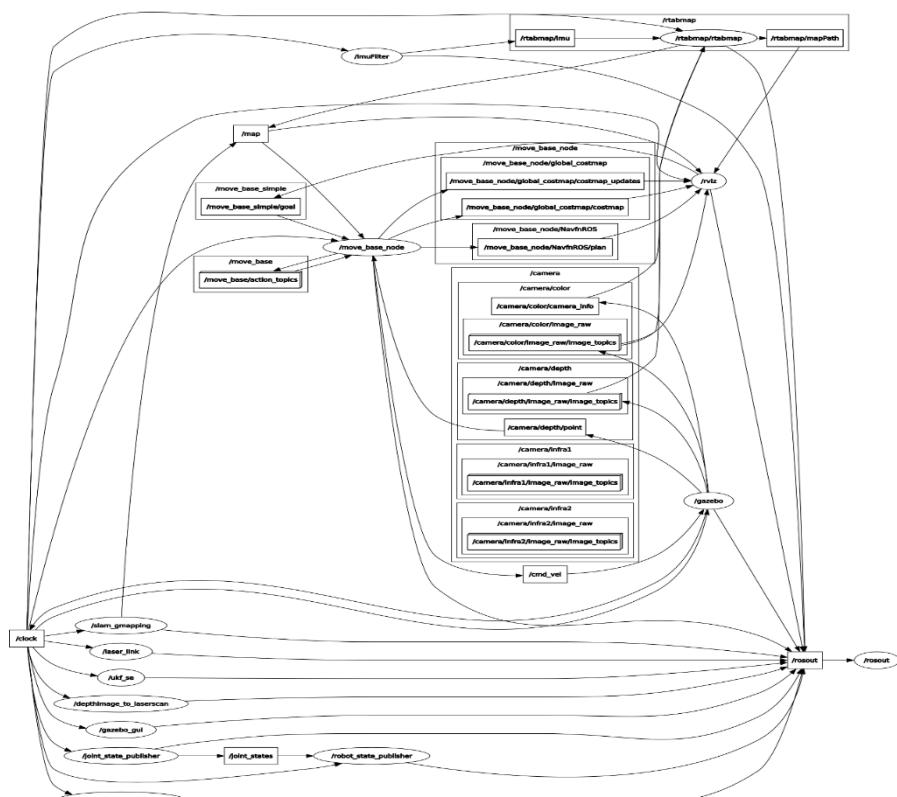
עבור שילוב ה-Gmapping עלתה שאלת חישוב המיפוי של הרובוט. שאלת שאלת נסעה עלי בחלוקת "ניתוח תוצאות" ו-"סיקום, מסקנות והוצאות להמשך".

עכט הרעיון הוא ששני האלגוריתמים יוכתבים לאותו Topic (map/) ובכך משפיעים על העבודה אחד של השני, יש צורך לפיכך למתאם בהםם שלא יפריעו אחד לשני.

הлокציה מתבצעת על ידי חיבור robot_localization בשיטת UKF, שפירתי עלייה ברקע התיאורטי. וכל אלגוריתם מיפוי מוסיף לכך את חלקו.

ובנוסף לשירותים מסווגים ROS, מימושו חבילת Move_base שאחראית לכל תחומי הניווט של הרובוט הכוללת Local Planner מסוג DWA שפירתי עליו ברקע התיאורטי.

מימוש ב-ROS : (rqt graph)



איור - 23 - rqt graph של מערכת ה-AI

באיור לעיל ניתן לראות את ה资质ים המתקשרים לצורך פעולה מערכת הבינה המלאכותית.

הערה : **rqt graph** המלא מופיע בחלק הנשפח של הדוח – נספח ב'.

מימוש הפרויקט זמין בעמוד ה-GitHub

<https://github.com/Bengal1/AI-for-Robot>

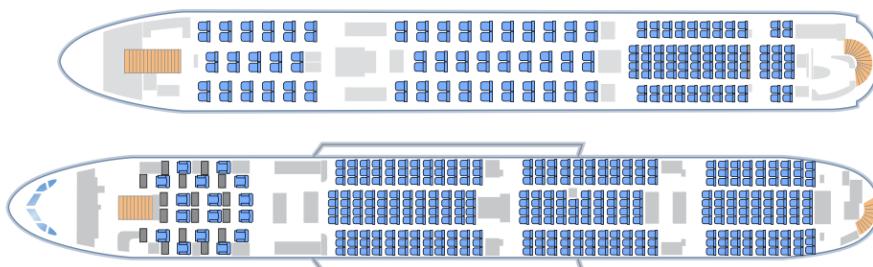


סימולציה

הפרויקט "רובוט לחיטוי מטוסים מקוונה" מורכב ממספר תתי-פרויקטים קטנים יותר, ומהסיבה הזאת לא ניתן לבצע מיפויים עם הרובוט האמיתי. על מנת לגשר על הפער הזאת יוצרת דגם של הרובוט, URDF, בעל תוכנות וזראה דומה לרובוט המתוכנן, ובסביבה המדמה את המთאר המתוכנן כדי לאמנו ולבדוק את מערכת ה-AI של הרובוט.

הסביבה הייעודית

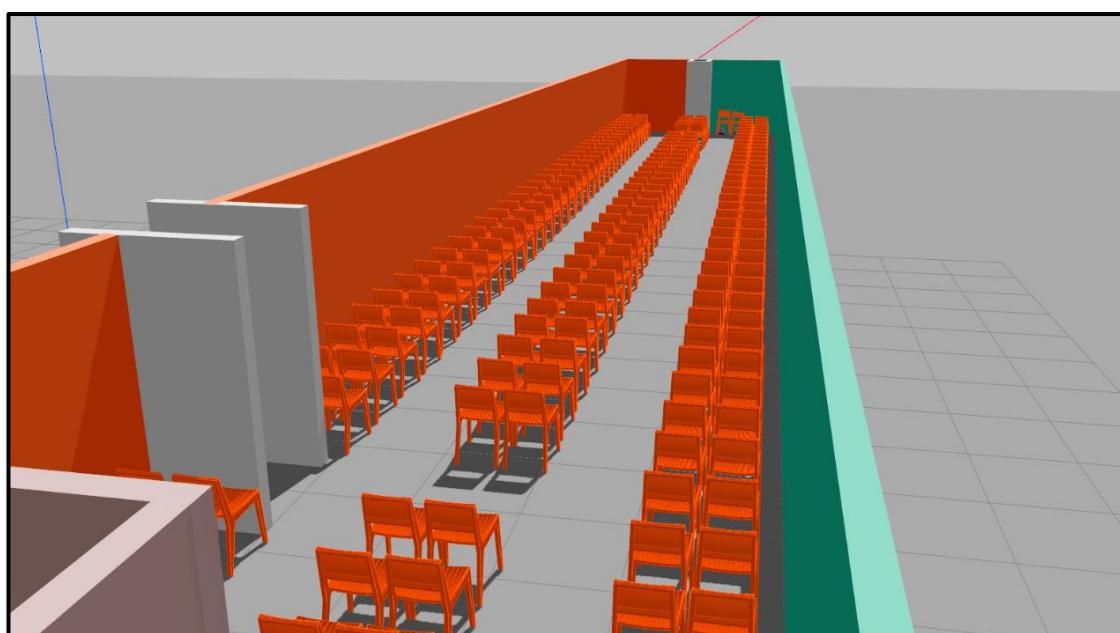
הרובוט מיועד לעבודה במתאר של מטוס נוסעים בחלל תא הנוסעים. באופן כללי תא הנוסעים מורכב משורות של כיסאות ומעברים בניהם. הרובוט צריך�� זהות את הcisאות כמכשורים ולנועם במעברים.



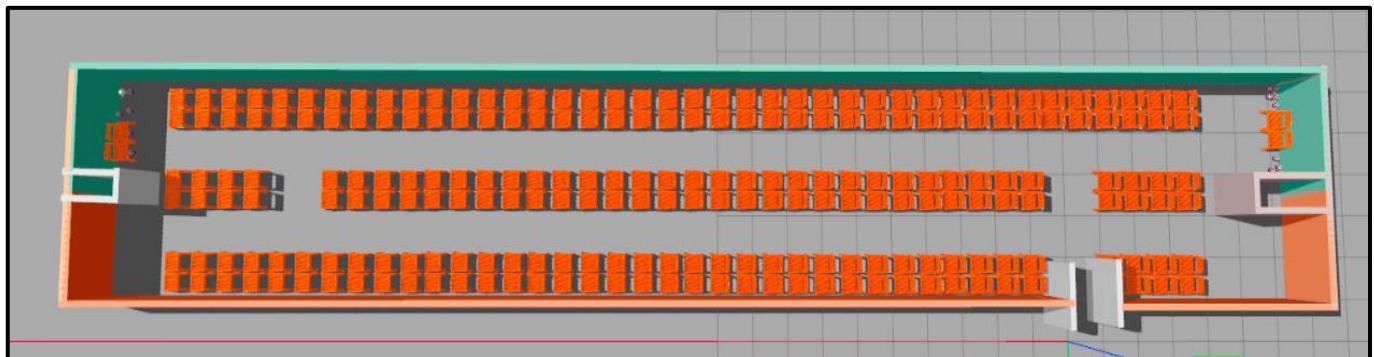
איור 24 - מודל תא נוסעים של המטוס נוסעים הגדול בעולם Airbus A380

סביבה הסימולציה

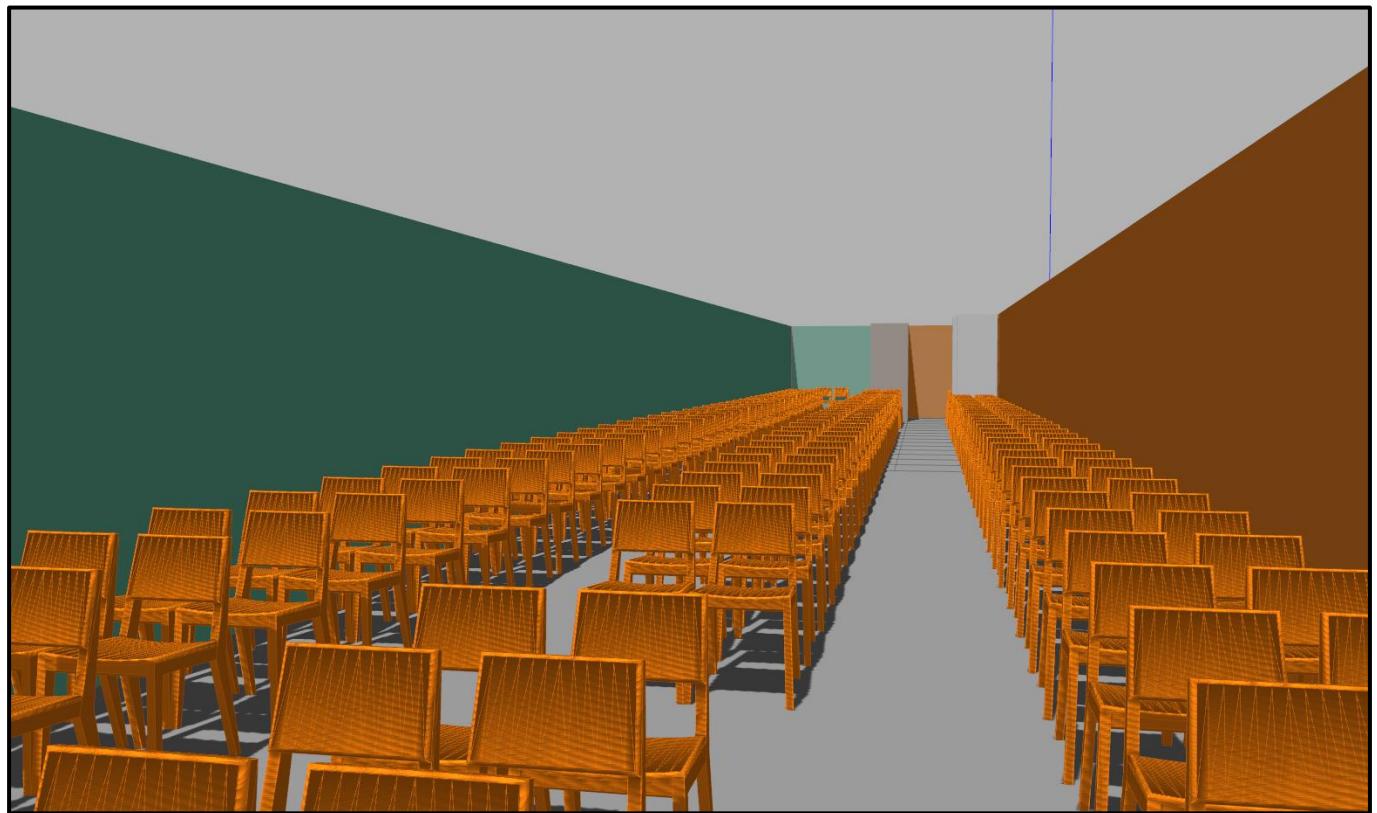
הסביבה הייעודית של הרובוט היא תא נוסעים של מטוס, וכן סביבת הסימולציה בנויה משורות של כיסאות עם המעברים ביניהם, סימולציה מוכנות לפי המכשורים שבחלל הנוסעים של המטוס. הסביבה נבנתה בעזרת תוכנת Gazebo תוך שימוש במודלים בעלי תכונות רלוונטיות הכוללים מימדים, אינרציה, Collision, פונקציונליות של מפרקים וSENSORS. בנוסף לכך הוכנסו לשכיבת הסימולציה פרמטרים נוספים לצורכי ביצורת, והם יפורטו בשלב ניתוח התוצאות.



איור 25 - תמונה של סביבת הסימולציה



איור 26 - סביבת הסימולציה ממבט על



איור 27 - סביבת הסימולציה מנקודת מבט פנימית

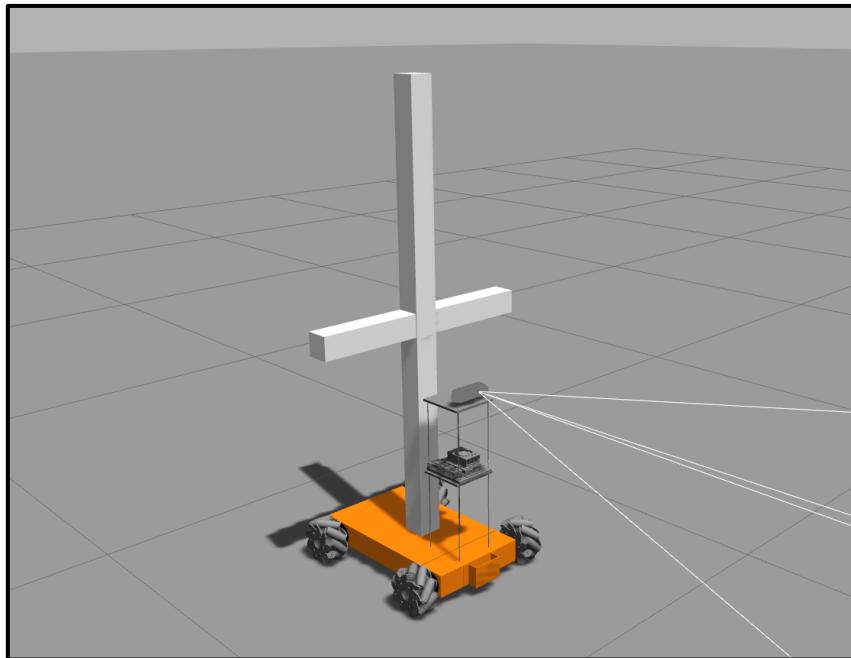
הערה: תМОונות נוספות של התוצרים מופיעות בגלריה בחלק "נספחים".



הרובוט

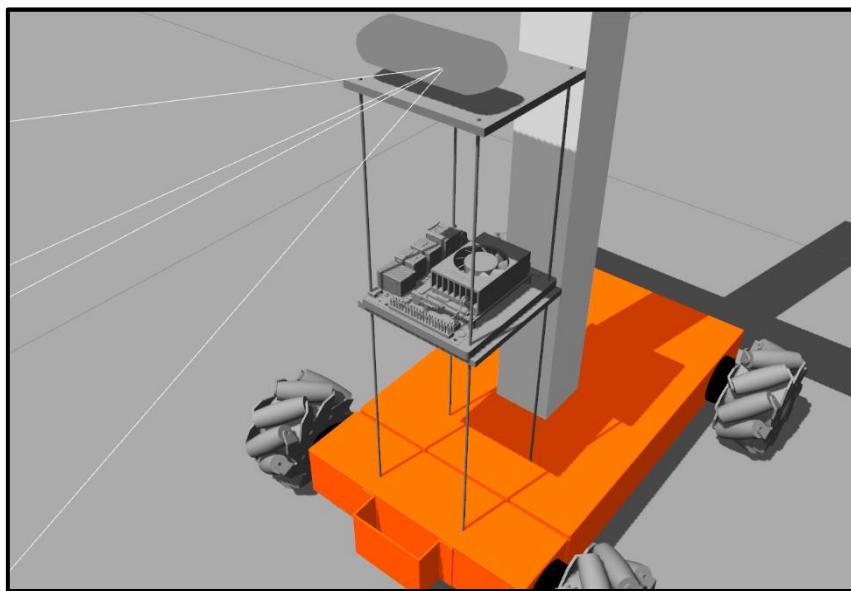
את הרובוט בניתי לפי התאור שסופק מוקבצת הרובוט. נשים לב שלרובוט גלגלי Mecanum. הרובוט מסוגל לבצע כל תנועה שגלגלים מסוגלים לבצע. באופן סימטרי הושפעת את זרועות הרובוט ללא LED או יכולת קיפול.

מימדי הרובוט נובעים ממשימת החיטוי. גובהו עולה במקצת על גובה הרכיסאות על מנת שיוכל לבצע את החיטוי כראוי. רוחבו גדול ממחצית רוחב המעבר על מנת שהתחיה חפיפה מספקת לחיטוי ללא פגיעה ביכולות התמרון של הרובוט. אורךו של הרובוט צריך להיות ארוך מספיק על מנת שהרובוט לא יתפרק כאשר הוא מבצע תנועה קדימה, ועם זאת לא ארוך מדי על מנת שיוכל להסתובב במקום באמצעות המעבר.



איור 28 - הרובוט בסימולציה

ניתן לראות למטה את מערכת הבינה המלאכותית המשומלצת הכוללת את המחשב של הרובוט (SBC) Nvidia Jetson Xavier NX והמצלמה של הרובוט iRealSense D435. נשים לב לגובה המצלמה, היביצועים המיטביים מתקבלים עבורי מצלמה שנמצאת מעל גובה אזורי היישיבה בכיסא (החלק המאוזן). במודל שלנו גובה המצלמה הוא כ- 42 ס"מ.



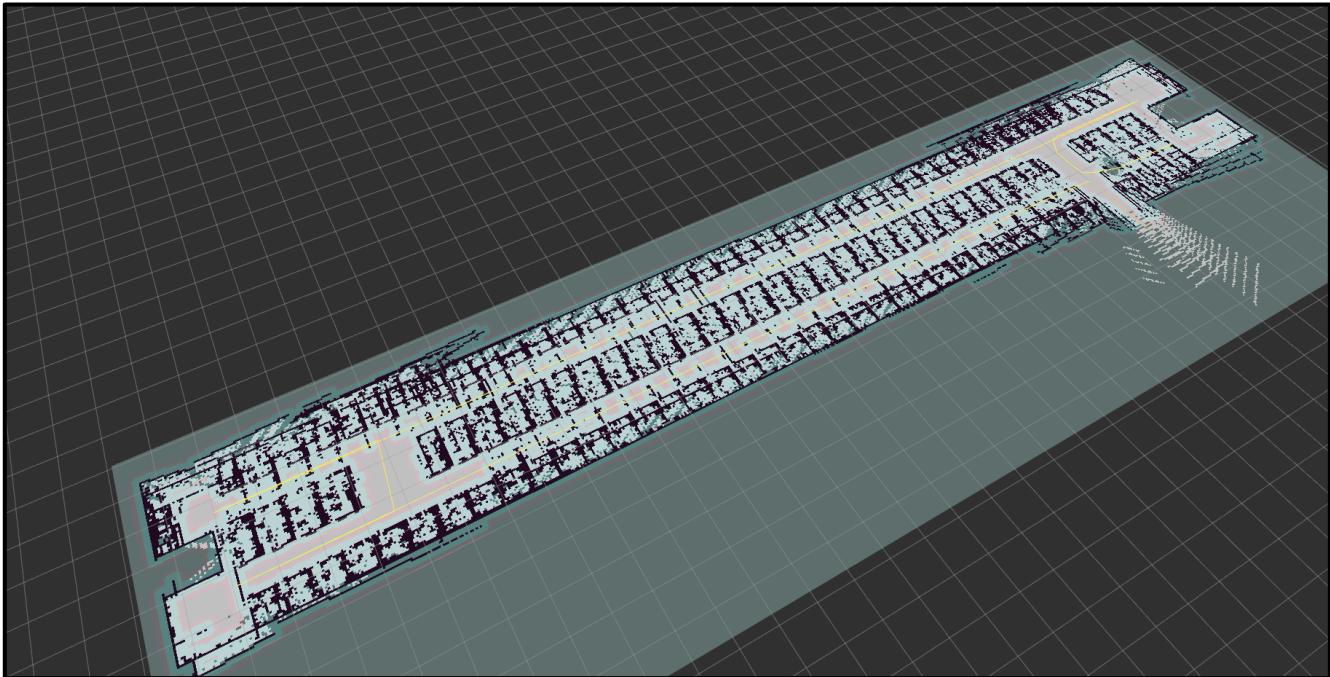
איור 29 - חומרת מערכת ה-AI בסימולציה



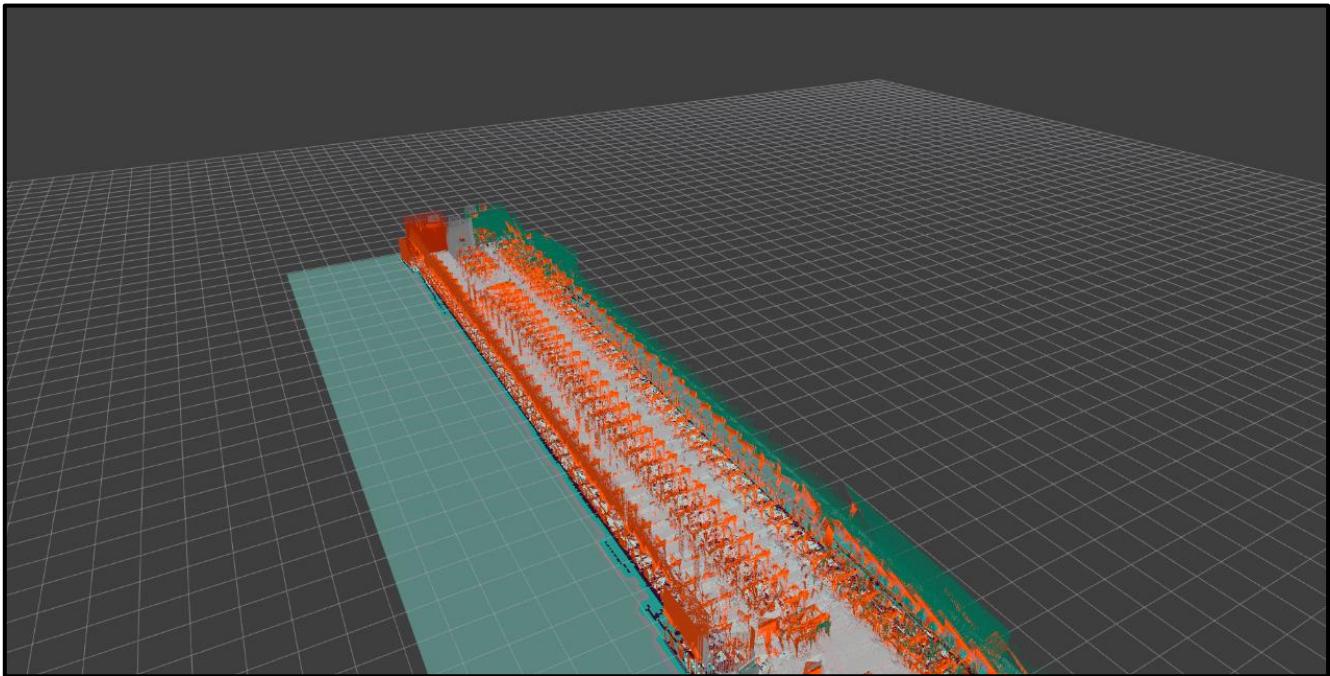
תוצאות הסימולציה

הסימולציה בוצעה בשני אופנים באופן ידני ובאופן חצי אוטונומי. הביצועים המיטביים התקבלו עבור הסימולציה שהתבצעה באופן ידני. תוצרת כל סימולציה הוא מפה תלת ממדית ומפה דו ממדית. בנוסף לשתי המפות אלה קיימים פריימרים (תמונות) הדגימה של אלגוריתם-h-RTAB-Map וマפת תשלום (Costmap) שם תוצרים פנימיים של האלגוריתמים לצרכי חישוב, מיפוי Lokvizition וכו'.

להלן תוצרי הסימולציה :



איור 30 - מיפוי דו ממדית של סביבת הסימולציה



איור 31 - מיפוי תלת ממדית של סביבת הסימולציה

המונות נספנות של התוצרים מופיעות בחלקים "ניתוח תוצאות" ו"בגלריה" בחלק "נספחים", והרחבה על תוצאות הסימולציה תתקיים בחלקים "ניתוח תוצאות" ו"סיכום ומסקנות".



ביצוע מיפוי על סביבה אמיתית

לאחר הצלחה של סיבוב הסימולציה, עולה הצורך לבדוק את יכולות המיפוי של הרובוט על סביבה אמיתית על מנת לבדוק את ביצועי הרובוט במתאר בו היא הסביבה אינה בהכרח מוגדרת היטב מבחינה גאומטרית.

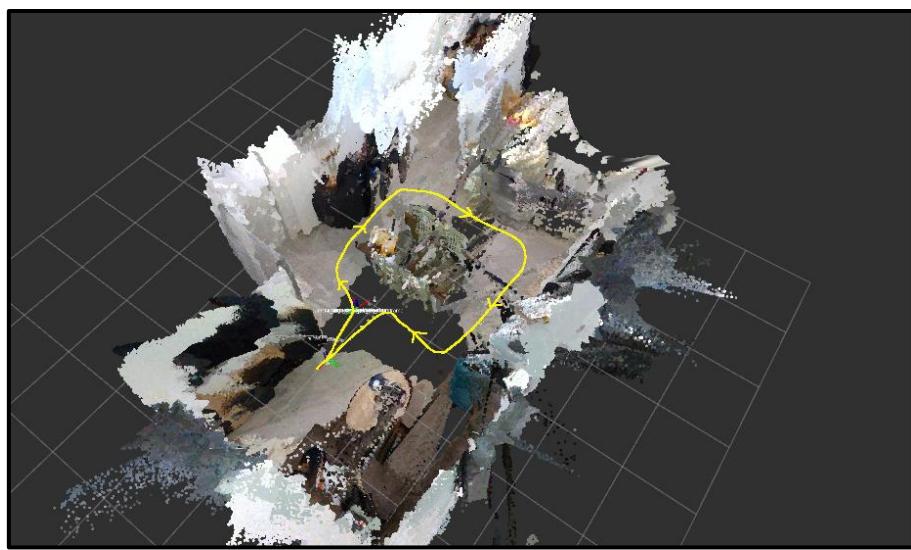
לשם כך הרכבתו מודל פסיבי המדמה את הרובוט, על בסיס כסא מחשב, ועליו כל החלקי החומרה הנחוצה, המכילה מוקבעת בקדמת הכסא, מקלדת ועכבר לשילטת ומסך 7 אינץ' לקבירה על המיפוי.



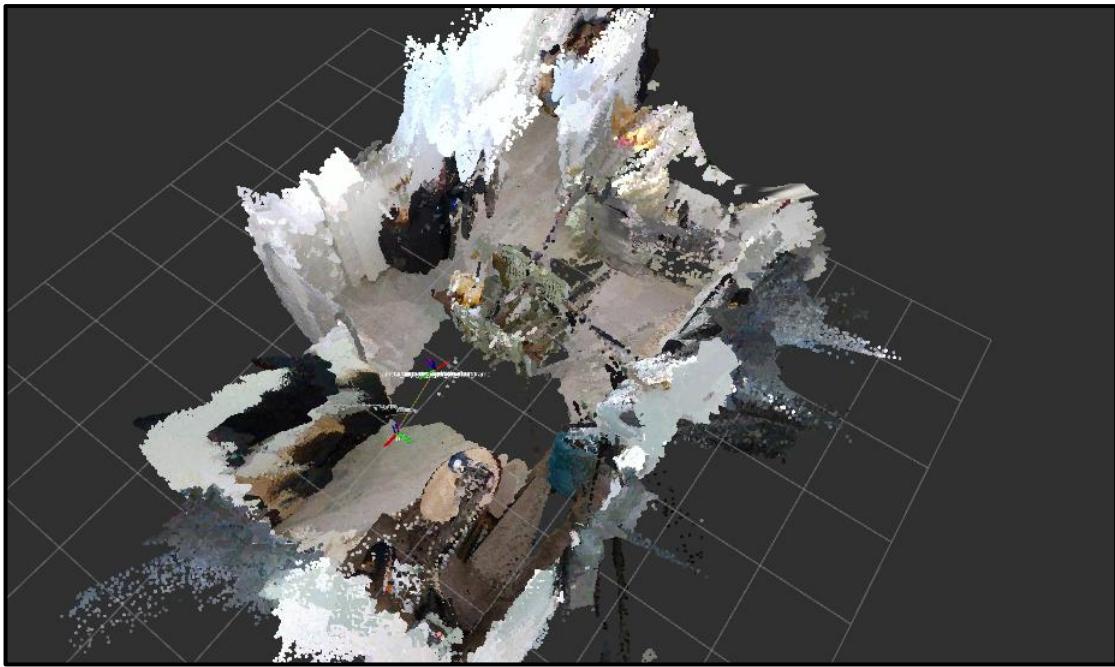
איור 32 - המודל הפסיבי (כסא) למיפוי סביבה אמיתית

את המיפוי ביצעתו בתוך החלל הראשי בדירה של המאופיין במעברים וחומרים שונים. בנוסף במסלול מעגלי מתousel מיפוי של טווחים שונים, ונitin לפיך להסיק את הטווח האופטימלי למיפוי תלת ממדים שמננו יגור המיפוי הדוו-מימדי.

ניתן לראות בתמונה למטה את המסלול שעברתי עם המודל הפסיבי (מסומן בצהוב) על מנת למפות את חלל הדירה.

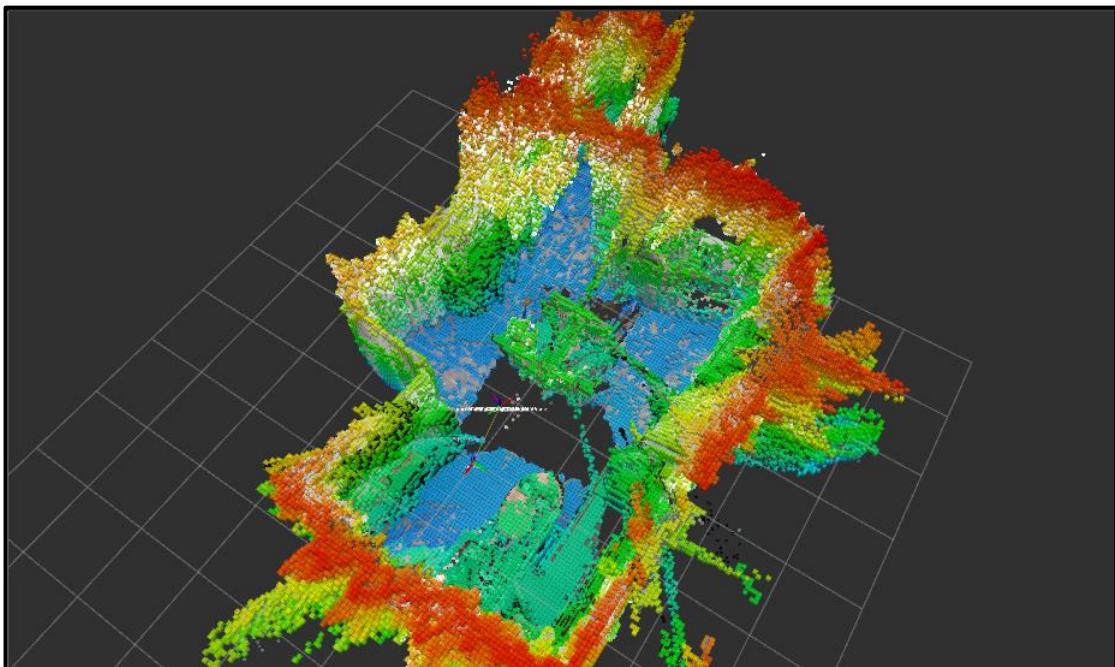


איור 33 - מפה תלת ממדית עם מסלול המיפוי



איור 34 - תמונה מייפוי תלת ממדית

מפת התפוצה שלעיל בניה מהצברות של נקודות ען (PointCloud), ניתן לראות בכמה מייפוי נקודות בודדות. נשים לב שפרט לאזור הרצפה התחלתי (והסופי) ה"רובוט" מיפה את כל החדר. ובנוסף נשים לב שהPointCloud מסודרים לפי הממדים והצבעים האמיטיים של חלל החדר.



איור 35 - תמונה Occupancy Map 3D

ניתן לראות שמפת התפוצה (Occupancy Map) שלעיל מקודדתocabים לפי ציר הגובה, ציר Z. נשים לב ששמפת התפוצה מחלקת את החדר לקוביות, ולפי הסריקות מקבלת החלטה האם הקובייה מהו חלל או מכשול. נראה שגובה הרצפה מקודד בכחול והגובה התפעולי של ה"רובוט" מקודד בירוק. החול מצחוב זהו גובה של 2 מטרים, גובה של ה"רובוט" אין כלל אינטראקטיבית בו.

הסרטונים של המיפויים זמינים בדף YouTube של הפרויקט:

<https://www.youtube.com/channel/UCwc1qD5aTrj4iaxQrpZu07w>



ניתוח תוצאות

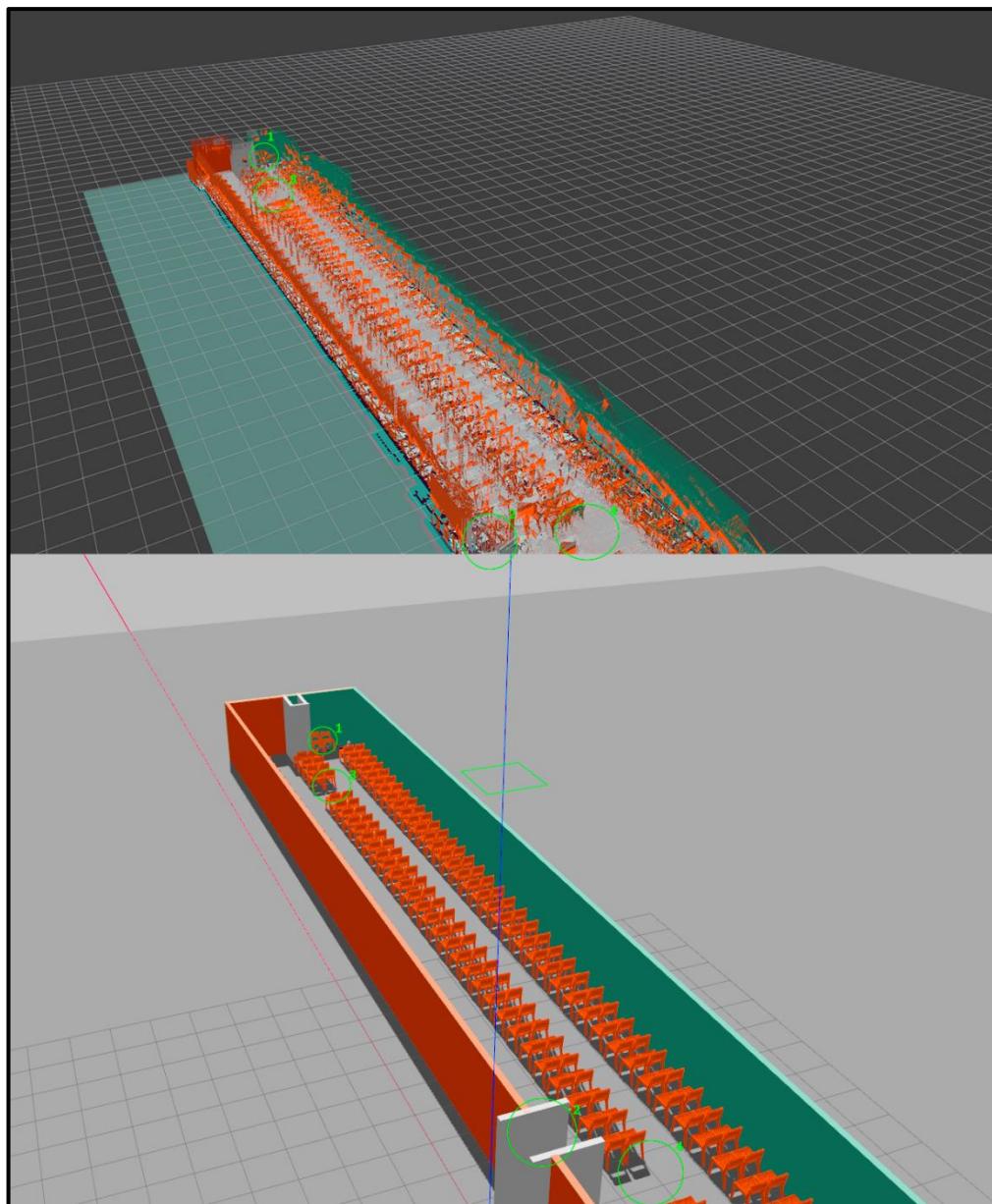
בזמן הסימולציה הרובוט מרכיב מספר מבנים נתונים, בהם מפה של תלת ממדית, מפה דו ממדית ובנוסח לשתי המפות אלה קיימים פרימיטים (תמונה) הדגימה של אלגוריתם-h-Map-RTAB (Costmap) שהם תוכרים פנימיים של האלגוריתמים לצרכי חישוב, מיפוי וлокציה וכן גם לבחינת הרובוט לצורכי תכנון, קינוח וככ'.

כאשר הרובוט מבצע את עבודתו בזורה טובה, הדבר מגיע לידי ביטוי במפות איקוניות אשר מייצגות נאמנה את המציאות. את תוצאות המיפוי ניתן לחלק לשני סוגים מיפויים:

מיפוי תלת ממדי (3D) אשר נדגם על ידי אלגוריתם-h-Map-RTAB, מיפוי תלת ממדי חשוב יותר לרשות, אך מהויה תמונה מצב מפורטים יותר עבור מפעיל הרובוט, וכן חשוב שהמיפוי התלת ממדי יבוצע באופן איקוני כך שנitinן להזות את הסביבה המומפה.

מיפוי דו ממדי (2D) אשר מ被执行 על ידי הטלחה של המיפוי התלת ממדית שモבצע על ידי אלגוריתם-h-Map-RTAB, ובנוסח על ידי אלגוריתם Gmapping (Particle Filter) אשר מ被执行 עבור נתונים הדגימה באופן שונה. חישוב המיפוי הדו ממדי היא הן עבור המפעיל והן עבור מערכת הניות של הרובוט, אשר מקבל החלטות ניוט על בסיס המפה הדו ממדית (וה-costmap) בגישה הדינמי (DWA).

מיפוי תלת-ממדי (3D)



איור 36 - השוואה בין מיפוי תלת ממדי של סביבת הסימולציה לסביבת הסימולציה

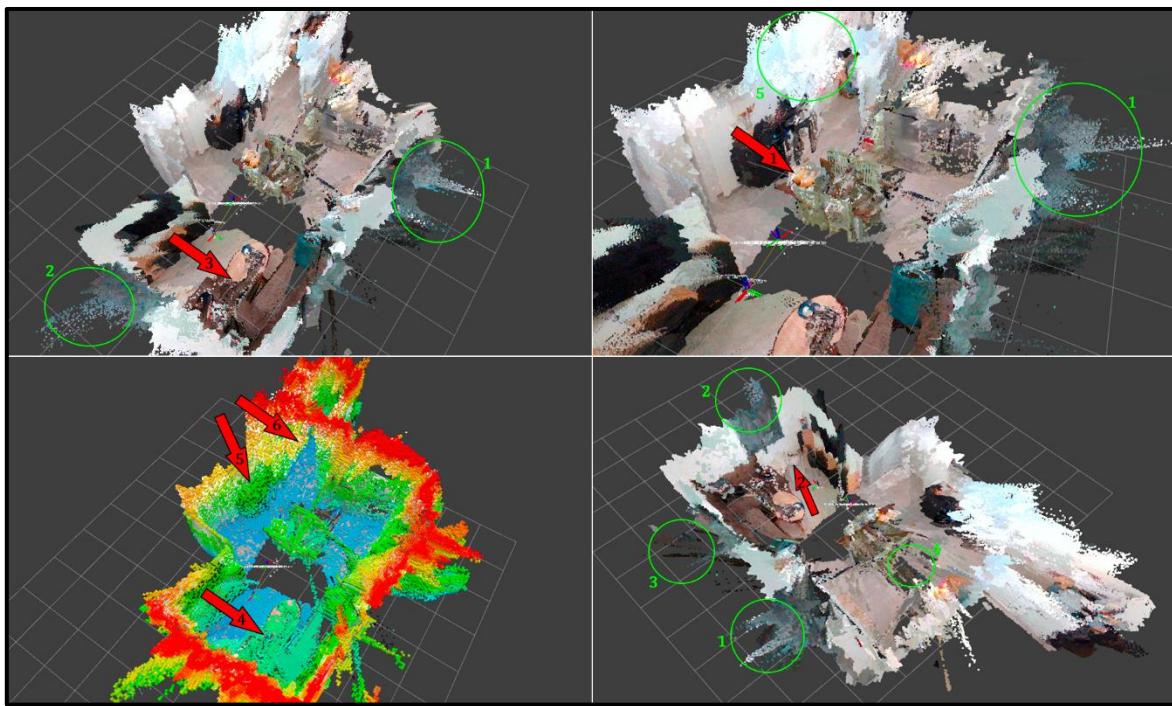


ניתן לראות לעיל השוואה בין המיפוי התלת מימדי של סביבת הסימולציה לסביבת הסימולציה עצמה. באירור שלעיל ניתן לראות ארבעה סימוני:

1. שני כיסאות בסוף המסדרון – סימון 1.
2. כניסה לחדר הסימולציה – סימון 2.
3. מעבר קדמי – סימון 3.
4. מעבר אחורי – סימון 4.

נסתכל בהשוואה ונראה שדימויו 1 ה被执行 בצורה טובת, אנחנו יכולים לזהות את הcisיות ואת מה שלדים, לעומת זאת ניתן לראות בסימון 2 שמיפוי קירות המעבר אין בודר באותו האופן, דבר שככל הנראה נובע מזמן השהייה אל מול הקירות ואבן השהייה, הרעש במיפוי מקטין תחילה מיפוי סביבת הרובוט אינו ממפה בטוחה אופטימלי את מעבר הכנסהויציאת של סביבת הסימולציה.

נשים לב שהמיפוי התלת מימדי הוא מיפוי רועש, בעיקר באזורי קירות סביבת הסימולציה ניתן לראות "ריוקשטים" ממיפוי cisיות. בנוסף ניתן לראות שבמיפוי התלת מימדי, הרעש במיפוי מקטין תחילה מיפוי המעברים הקדמי והאחורי (סימון 3 וסימון 4), אך עדין מספק מיפוי מעברים יחסית איקוטי כך שניתן לרובוט לעبور בצורה בטוחה.



איור 37 - מיפוי תלת מימדי מכמה זוויות והציגות עם סימוני

עליל ניתן לראות את המיפוי שהבוצע בדירתית מכמה זוויות ובנוסף Occupancy Map שמקודדת בצבעים לפי ציר Z (גובה). נשים לב לסימונים על המפות כאשר העיגולים הירוקים (5-1) מסמנים רעש חריג והחיצים האדומים (6-1) מסמנים נק' עניין לניטוח התוצאות.

ראשית נשים לב שימוש חלקיים דמיי מראה מייצרים רעש רב במפה, ניתן לשים לב רוב הרעש במפה נמצא באזורי הגבוחים של חלונות הדירה סימונים ירוקים מספרים 1-3, וכן באופן דומה באזורי התנור והמיקרו גם כן עם שטח פנים דמיי מראה, סימון 3 (הסימון הוא משותף לרעש מהחלון ולרעש מהתנור והמיקרו).

כמו כן אזור רעש נוסף נמצא בין השולחן למקדר (מסומן בתמונה הימנית תחתונה – סימון ירוק מס' 4) דבר שנובע ככל הנראה מצורת הסורגים של משענת הכסא שגורם ל"ריוקשטים" הנקודה המדוברת.

נראה את סימוני החיצים האדומים מס' 1 ומספר 2, נשים לב כי המיפוי הה被执行 כהכלחה וחץ מס' 1 ניתן לראות את השולחן (רעש מעט) עם פריטים עלייו, ובחר מספר שתים ניתן לראות את שואף האבק הרובוטי באופן יחסית ברור.

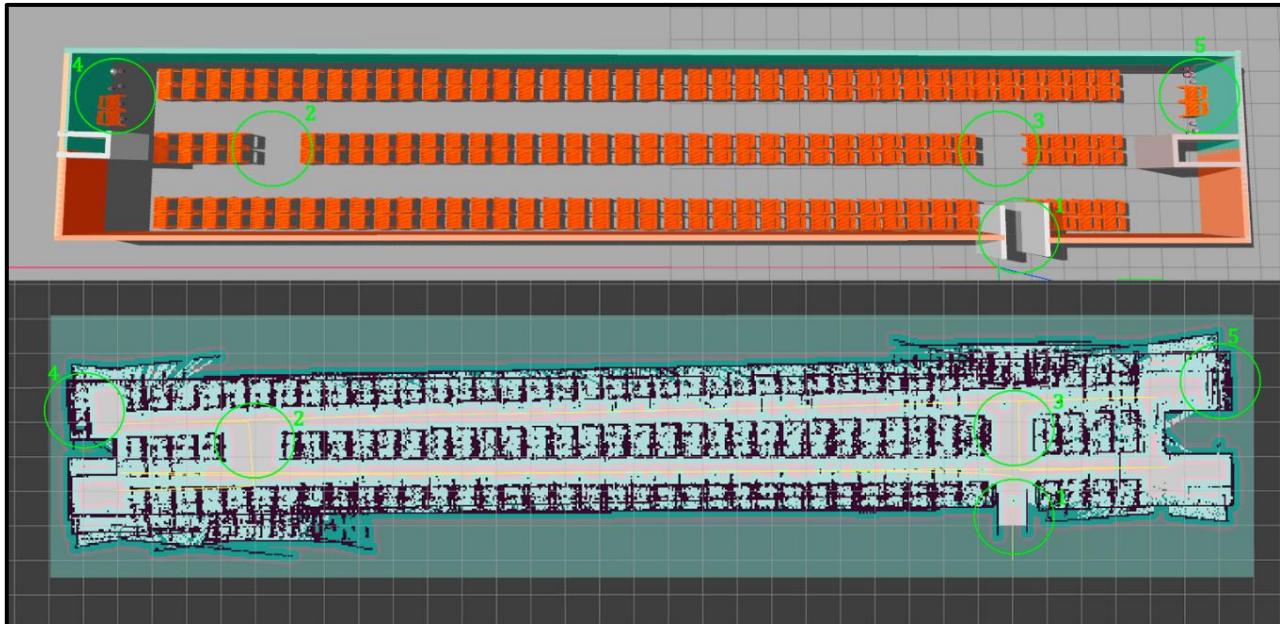
סימוני החיצים האדומים 3 ו- 4 הן אותה נקודה, אחת במיפוי 3 והשנייה ב-Map 2 בהתאם. נראה שבאזור הגובה שמקודד בירוק המיפוי הוא מיטבי, וכן אפשר לראות בנקודה המדוורת את השולחן בסלון באופן ברור מאוד. ובאותו האופן גם בנקודה 5, את פינת הקפה.

חץ אדום מס' 6 מסמן את המעבר למזווהה שמופה באופן חלק פרט לאיזור מעליו שמקודד באדום והרובוט אינו מגעה עימיו באינטראקציה ישירה, ובוא רעש צפוף יחסית של PointCloud, נתיחס לכך בחלק המיפוי הדו מימי.



אצין בנוסח סגירת הלולאה (Loop Closure) ופקד היבר וכפי שניתן להראות לא קיים במיפוי Wheel Odometry שאמור לשפר את המיפוי בaczera משמעותית. אך יכול להתבצע רק drift, וזה עוד ללא שימוש-b- drift. עם הרובוט האמיתי אשר מכיל בקר שסופר את ה-step של המנוונים, יוכל לדוח חזרה למערכת הבינה המלאכותית.

מיפוי דו-מימדי (2D)



איור 38 - השוואת בין מיפוי דו מימדי של סביבת הסימולציה לסביבת הסימולציה

ניתן לראות לעיל השוואת בין המיפוי הדו מימי של סביבת הסימולציה לסביבת הסימולציה עצמה. באיזור שלעליל ניתן לראות חמייה סימוניים:

- | | |
|---|----|
| כניתה לחיל הסימולציה – סימון 1. | .1 |
| מעבר קדמי – סימון 2. | .2 |
| מעבר אחורי – סימון 3. | .3 |
| מכשולים בחלקו הקדמי של המסדרון העליון – סימון 4. | .4 |
| מכשולים בחלקו האחורי של המסדרון העליון – סימון 5. | .5 |

כאשר מסתכלים על חלל הכניסה לשבית הסימולציה (סימנו 1) ניתן לראות כי המיפוי מדויק בזורה כזואת שניתן לראות את הדירוג בו שני קירות הכניסה (הקיר הימני גבוה מעט מהشمالي – עשה לצורכי ביקורת). בסופו ניתן לראות את המעברים הקדמי והאחרי (סימנום 2 ו-3) בזורה מאד ברורה, וכן גם את יחס הגדים בהםם (המעבר הקדמי גדול פי 1.3 ~ – עשה לצורכי ביקורת).

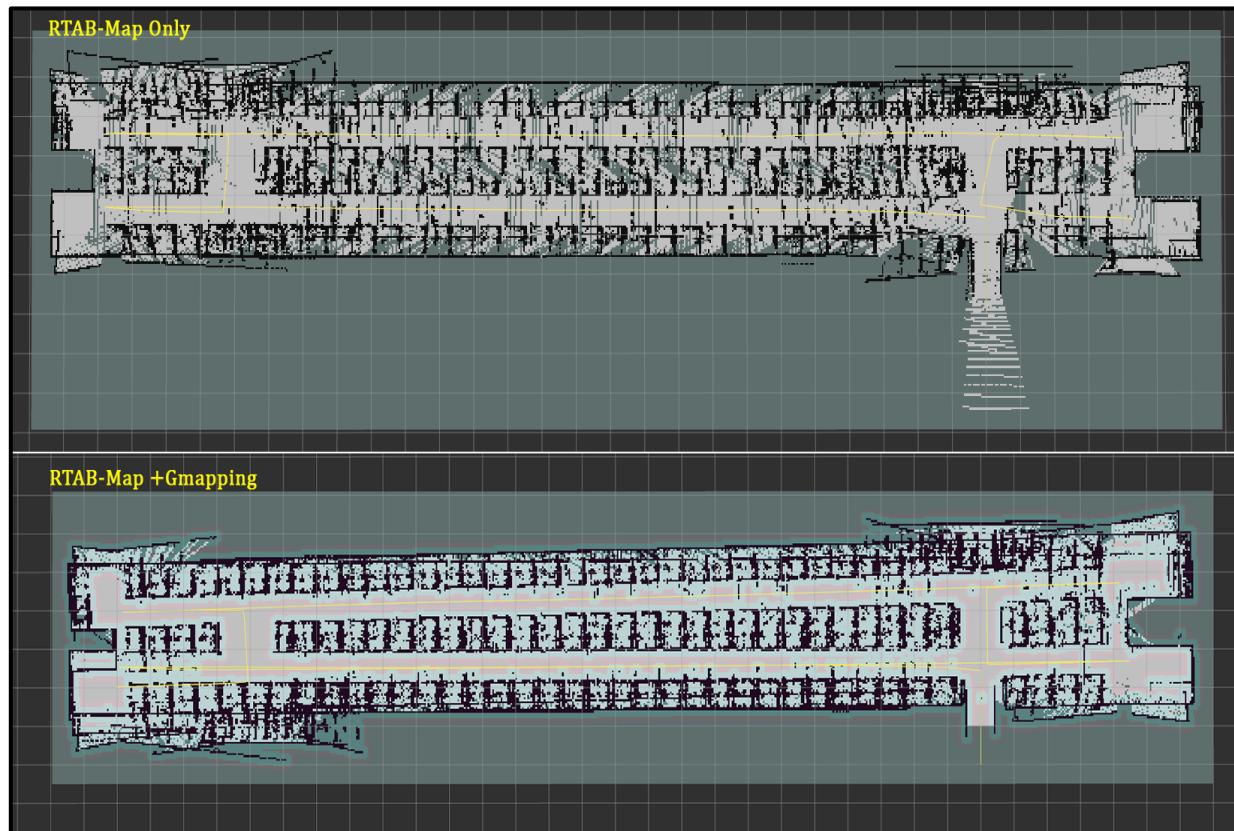
בנוסף לכך נשים לסייעים במספר 4 ומספר 5, בחלק הזה הצבתי מכשולים שימושיים לשבירת שיגורה עבר הפריימרים של RTAB-Map ועבור ביקורת. ניתן לראות שהרובוט מיפה אותם בצורה טובה מאוד, כך שניתנו להיזהר מהם על ידי המפה הדור-מידית בלבד.

עם זאת ניתן לשים לב כי המטדרוגנט רועשים מעט, דבר שככל הנראה נובע מכך שהתוצאות בשילוב אופן המיפוי של אלגוריתם-h-RTAB-Map, ראיינו זאת בסביבה אמיתית עם ביציאות המטבח שלו (סימון יירוק מספר 4 באירוע מס' 37). ובנוסף קיים עיון כל של המטדרון התחנו באזורי הקדמי שלו (השמאלי).

בשלב הבא נשווה בין יכולות המיפוי של הרובוט כאשר בחלק הראשון המיפוי של סביבת הסימולציה מתבצע רק על RTAB-Map + Gmapping, ובחלק השני המיפוי מתבצע על ידי RTAB-Map + Gmapping. מכיוון שאלגוריתם RTAB-Map מצליח מאוד במיפוי תלת מימדי ורשות מאוד במיפוי דו-ממדדי הוספה אלגוריתם Gmapping שהוא מסוג Particle Filter (ראה רקע תאורטי), יסיע לרובוט באופן בלתי תלוי באlgorigthm ה-RTAB-Map ויספח עוד מידע לקבالت החלטה במיפוי דו-ממדדי עבור מ鏘ולים בחלל.



להלן השוואת מיפוי חלל הסימולציה (MAPPING DO MIMDI), כאשר המפה העליונה שייכת למיפוי שנעשה בעזרת RTAB-Map בלבד והמפה התחתונה הורכבה על ידי RTAB-Map + Gmapping.



איור 39 – השוואת בין מיפוי דו מימדי של סביבת הסימולציה RTAB-Map ל- RTAB-Map + Gmapping

ראשית כל, ניתן לראות במבט על חתוֹף כי המכשולים במיפוי המשולב ברורים ממשוערת יותר מאשר המכשולים במיפוי RTAB-Map בלבד. את בהירות המפה התחתונה ניתן לנכס לעובדה שולב אלגוריתם מיפוי Gmapping, לאלגוריתם RTAB-Map, ובנוסף ניתן לראות שהשילוב מנקה את הרעש החיצוני (מהוזע לחיל הסימולציה).

ניתן לראות שאלגוריתם RTAB-Map מבודד מפה היטב את חלל הסימולציה, אך באופן מאוד רועש, נשים לב לחוליות (Graffiti) המוגברת באזורי המסדרונות, לעומת המיפוי המשולב שלא חף מרעש המסדרונות, אך נקי מחשיבות מהראשו.

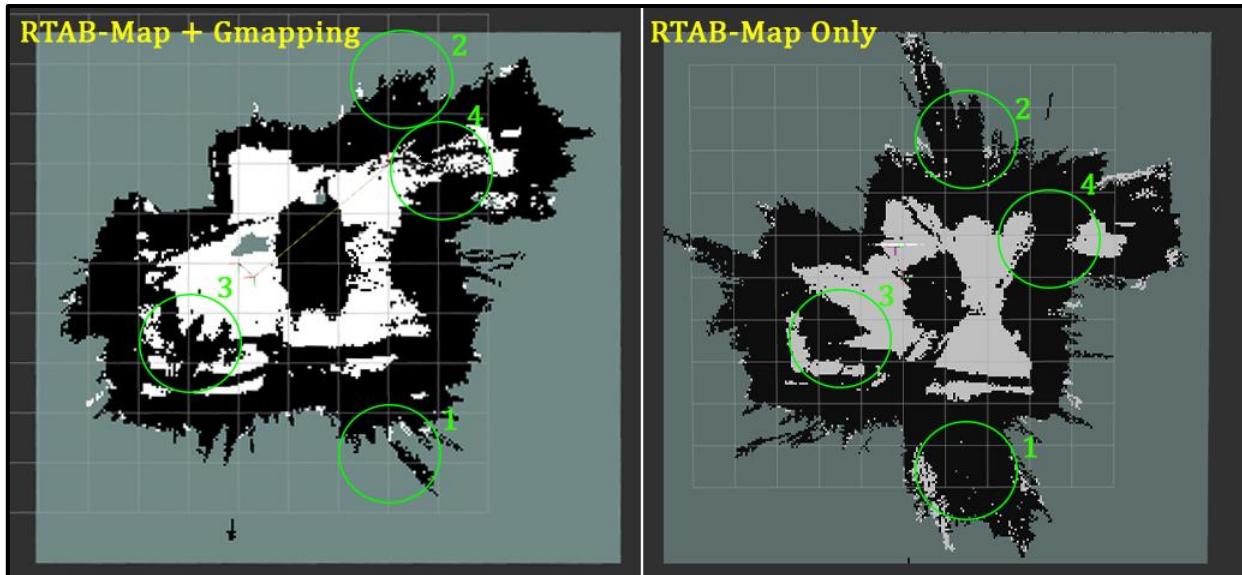
ניתן לראות במיפוי RTAB-Map בלבד חלקים חסריים במפה, בעיקר עברו אזורים שאין בהם PointClouds. דבר זה נובע ככל הנראה מאופן המיפוי של RTAB-Map, אשר מבצע מיפוי תלת מימדי (3D) ולאחר מכן מצבע הטליה על המשיר לצירית מפה דו מימדית. נראה כי גם במיפוי התלת מימדי באירועים 34 ו-37 ניתן לראות את התופעה הזאת. אך כאשר משלבים את אלגוריתם Gmapping, אשר משפיעה רק במיפוי הדו מימדי ניתן לראות ירידת משמעותית בכמות החורמים, כמעט ואינו חורם במיפוי המשולב, וגם אלא שקיים קטנים מאוד.

נשים לב שלא קיים הבדל בגודל בזיהוי המכשולים שבקטוזות המסדרון העליון עבור שני המיפויים ובשני המיפויים ניתן לראות בבירור את שני המסדרונות. אך נשים לב לנוטני הביקורת, ניתן לראות במפה העליונה שלא רואים את דירוג הקירות בחיל הכנסה, לעומת המפה התחתונה של המיפוי המשולב שבו כן רואים (הקיר הימני גבוה מעט מהشمאלית – נעשה לצורכי ביקורת). ובנוסף גבולות המעבר הקדמי במיפוי הבודד (רק RTAB-Map) לא ברורים לעומת המיפוי המשולב בו ניתן לראות את היחס בין גדרי המעברים המעבר הקדמי גדול פי ~ 1.3 – ~ נעשה לצורכי ביקורת).

ובאופן כללי, ניתן לראות כי עברו המיפוי המשולב (RTAB-Map + Gmapping), קיבל מפה דו מימדי איכותית יותר של סביבת הסימולציה, הן מבחינה כמותית: זיהוי מכשולים, יחסים בין גדלים ומיקומים, והן מבחינה איכותית: רעש שאזורים קריטיים, סחיפת המפה של Odometry ובhartoot המפה.



להלן השוואת של מיפוי חלל הדירה שלי (מיפוי דו מימדי על סביבה אמיתית), כאשר המפה הימנית שייכת למיפויalone : Gmapping RTAB-Map בלבד והמפה השמאלית הורכבה על ידי RTAB-Map



איור 40 - השוואת בין מיפוי דו מימדי של סביבה אמיתית ל- RTAB-Map + Gmapping

באיור ההשוואתי שלעיל ישנים ארבעה סימונים :

1. רعش מחלון המטבח – סימון 1.
2. רعش מפינת הקפה (כמה משטחים עם יכולת החזרה) – סימון 2.
3. הסלון – סימון 3.
4. המעבר למזווה – סימון 4.

ניתן לראות הבדלים מאוד ברורים בין שני המיפויים, ראשית נתיחס לסימונים שבאיור לעיל.

נשים לב שבסימונים מס' 1 ומס' 2 אשר מביעים רעשים ממוקמים בעITIES שנובעים ממשטחים דמיויי מראה, נשים לב שאלגוריתם ה-Gmapping מרסן את הרעש שמתתקבל על ידי אלגוריתם ה-RTAB-Map בלבד, ונinanן לראות ירידת משמעותית ברעש באזורי הסימונים הללו בהשוואה למיפוי שהבוצע על ידי אלגוריתם ה-RTAB-Map בלבד.

כאשר נסתכל על סימון 3 נראה כי אזור הסלון גם כן רועש משמעותית עבור המיפוי הדו מימדי של אלגוריתם RTAB-Map, לעומת המיפוי הדו מימדי המשולב. במיפוי המשולב ניתן לראות את צורתה ה-ה' של השפה בעוד במיפוי שנעשה על ה-RTAB-Map בלבד הטרורה מתייעצת. ובאותו האופן השולחן במרכז המפה (הכתם השחור במרכז) המפות באייר (40) במיפוי הדו מימדי של ה-RTAB-Map הצורה המתתקבלת של השולחן היא אלייפיטית/אמורפית, לעומת החלטה שהתקבלת עבור מיקום המכשול איננה הייתה החלטה טוביה לעומך המיפוי המשולב, אשר ניתן לראות כי צורתה השולחן קרובה מאוד למלבן וגדולה יותר, דבר שמעיד על החלטה טוביה לגבי מיקום מכשולים של שילוב האלגוריתמים.

סימון מס' 4 הוא אזור המעבר למזווה, אזור שעלה במיפוי החלטת מימדי בסימון מס' 5 באיו 37. ככל הנראה בגלל קרבת הקירות הלבנים אלגוריתם RTAB-Map מיציר רעש רב באזורי זה. ניתן לראות זאת במיפוי הדו מימדי של אלגוריתם RTAB-Map בלבד באופן ברור, ככל הנראה דבר שנובע מاؤון המיפוי הדו-מימדי של אלגוריתם RTAB-Map, וכן גם במיפוי של האלגוריתם המשולב באופן קל יותר. נשים לב שעבור המיפוי הבזבז RTAB-Map בלבד המעביר ממש חסום, לעומת ענטת הרעש שניתן לראות במיפוי הדו מימדי של שילוב האלגוריתמים. דבר אשר מראה את העליונות של שילוב אלגוריתמי המיפוי על פני אלגוריתם ה-RTAB-Map בלבד.

בחלק ניתוח התוצאות בchner איקוטית וכמותית את מיפוי הרובוט בסביבת סימולציה ובסביבה אמיתית, ובנוסף הראננו את עליונות שילוב אלגוריתמי המיפוי RTAB-Map ו-Gmapping על פני אלגוריתם ה-RTAB-Map בלבד.



סיכום מסקנות והצעות להמשך

סיכום

בפרויקט זהה בנוינו מערכת בינה מלאכותית עבור רובוט לחיטוי מטוסים מקורונה. המערכת משתמשת בטכניקות מיפוי מהרינה הגדולה ביותר הזרינית למפתח הפרטיל ולקהיל הרחב. אלגוריתם המיפוי RTAB-Map שמיושם על ידי מחלקה של Intel, והוכח את עצמו מספר רב של פעמים במיפוי RGB-D, Gmapping שנחשב לאחד מיישומי-hParticle Filter והזמינים ביוטר, ובנוסף מושך בחבילה מובנת ב-Extended Kalman Filter. המערכת מבצעת לוקליזציה על ידי Unscented Kalman Filter אשר ייעילו אל מול Kalman Filter שנמצא בשימוש נרחב.

בפרויקט זהה מיפויו סביבת סימולציה שנבנתה ב-Gazebo על מנת לדמות סביבת חלל תא נסעים במטוס, וסביבה אמיתית (יראה), וכל זה על מנת להראות התכונות והפונקציונליות של המערכת. ובנוסף הוכחנו שעבור קומפיקורציית החומרה שלנו שילוב אלגוריתם Gmapping מיעיל את המיפוי שמאבע אלגוריתם RTAB-Map.

נראה כי דרישות הפרויקט עמדו במלואן, אך עם זאת ישנים עוד דברים שניתן לשפר ולפתח. באופן אידיאלי נרצה שתוצרי המפות של הרובוט, המפה ההתלת מימדיית והמפה הדו מימדיית תהינה נקיות מרעשים מיותרים. בנוסף נרצה שהרובוט ינותן באופן אוטונומי מלא בחול המטוס החל מרגע ההפעלה יסורך את כל חלל תא הנוסעים ויחזור לנקודות ההתחלת באופן עצמאי.

כמו כן, שלב החיבור של מכלול הפרויקטים לא קרה עדין בו נדע את היכולות והפונקציונליות האמיתית של הרובוט כיחידה.

רשימת תוכרי הפרויקט		
פירוט	תיאור	
,URDF ,librealsense ,ROS : רכישת כל הדוד לעיר שאר התמירים .Linux,gazebo ,Xacro ,SDF	למוד המידע התייאורי והטכני.	Administration
single board מצלמת עמוק ברוולチיה מספקת וחבילה تعمل מספקת, .(SBC) comuter	רכישת מצלמת חומרה.	
חיבור הפני בין המחשב למצלמה Real Time ,קיינוגים, התקנה וכיויל של gazebo plugin .librealsense	IMPLEMENTATION בין המצלמה למחשב.	
בנייה ורבות בפורמט אחד (URDF) למיפוי סימולציה, בניית -world gazebo וקיינוגים.	IMPLEMENTATION URDF של הרובוט והסימולציה.	SLAM
,robot _localization Node ,Gmapping Node,RTAB-Map Node ,depthimage_to_laserscan	IMPLEMENTATION SLAM וממשק.	
IMPLEMENTATION global planner ,local planner ,Move_base node ,קיינוגים וטסטים.	IMPLEMENTATION ניוט.	Navigation
איחוד התוכרים, וסימולץ הרובוט בהתאם תא נסעים של מטוס.	איחוד התוכרים לכדי סימולציה.	Simulation

טבלה 2 - טבלת תוכרים



מסקנות

במסגרת ביצוע הפרויקט "מערכת בינה מלאכותית עבור רובוט" נוסו מספר אפשרויות כפתרונות עבור דרישות המערכת התואם לאיולוי הזמן והמשאים שניתנו. חלק מהאפשרויות לא כלחו עקב חוסר בזמן, וחלק אחר לא היו מתאימות או לחילופין לא טובות מספיק או לא עומדות בקריטריונים שקבעו לטובת מערכת הבינה המלאכותית. תוצאה לכך עלו מספר מסקנות שישיעו להמשך הפרויקט הגדול, וכן עבור שחזור הפרויקט הנוכחי "מערכת בינה מלאכותית עבור רובוט".

המסקנה הראשונה הבולטת ביותר היא שמיומוש שני אלגוריתמי מייפוי יכולים להבא לשיפור המיפוי של המערכת. דבר שלא מוביל מכיוון שני האלגוריתמים מkładים כיסיה מאותו מקור, מצלמת העומק והמייפוי הדור מידי עשוי להתכנס לטוב מביין השניים (אלגוריתם דומיננטי) או למוצע של שנייהם (אלגוריתם חלש שהורס את המייפוי לאלגוריתם החזק).

ל-Xavier NX Nvidia אפשרות לפעול בחישכון הספקי, [W] 10, בכירית מוחדר או בהספק מלא, [W] 15. על מנת שייהה אפשר להשתמש בסוללות "חלשות" יותר כמו קור אנרגיה עבור הרובוט. מהפרויקט הזה נובע באופן חד משמעי שלא ניתן ומעבר לכך-Nvidia Xavier, למרות היותו מחשב חזק בעל כוח חישובי גבוה נמצאת באופן שיגרתי תחת מגבלה, ושימוש במערכת הבינה המלאכותית צורך משבאים רבים וכוח חישוב רב. בהפעלת המערכת לזמן ממושךணספ לשימוש במקבץ 6 ליבות יש לשחרר את המוגבלות של המחשב על ידי שימוש ב:

```
$ sudo /usr/bin/jetson_clock
```

בתחלת הפרויקט עלתה השאלה האם כדאי להשתמש במצלמת עומק או ב-2D LiDAR, כחומרה שדוגמת את הסביבה. ככל אחד מההתקנים היתרונות והחסרונות שלו, ומה שהוביל בסוף להחלטה לבחור במצלמת העומק (RGB-D), הוא יכולת הניזוט של המפעיל, החלטה שהתרברה עדיפה יותר לאור העובדה שנitin לדמות סריקות לייזר מתחומות עומק. המפעיל שאינו רואה את הרובוט צריך לדעת את מצבו ומיקומו ביחס למפה, ומצלמת העומק כדי שנitin לראות מספקת את תומנת המצב בקרה המיטבית עבור בן-אדם (פעועל).

מכיוון שלפרויקט הגדול "רובוט לחיטוי מטושים מקורונה" מספר צוותים, הייתה מידה מסוימת של חוסר תיאום ומשימות של שילוב הפרויקטים הקטנים לא התממשו. למשל הממשק של מערכת הבינה המלאכותית עם הרובוט הוא דרך הבקר, אך מכיוון שהוא שוני מימי פROYקטים נפרדים חלק התוכנה של הבקר לא מושם. לכן יש לבצע תיאום בין הצוותים, במידה זאת שתאפשר את עבודה כלל יחידות הרובוט, ובנוספ את חלקים המשקימים בין היחידות.

הצעות להמשך

לפי שעה לא קיים קישור תוכני בין הבקר למערכת הבינה המלאכותית. נוסף לכך שפוקודות ניוט צריכות להציג לשאר יחידות הרובוט, מערכת הבינה המלאכותית צריכה לקבל Wheel Odometry מהబקר כדי לבצע מייפוי וולוקלייזציה בקרה מיטבית. לכן, יש לבנות את ממשק הבקר ב-ROS. לשם כך קיימות ספריות שקילו את העבודה, הפעולה דורשת עבודה מסוימת עם מחשב הרובוט, Nvidia Jetson NX ובקר הרובוט Arduino Mega 2560.

כפי שנטענו לפני, באופן אופטימלי נרצה שהרובוט ינוט באופן תקין הפעלה יסורך את כל תא הנוסעים ויחזר לנקודת ההתחלה באופן עצמאי. לשם כך למערכת הבינה המלאכותית צריך להיות קיימים Global Planner ו-Local Planner, Global Planner מתקן את דרך הרובוט לפי נתיבים פנויים כל אינטראול זמן לפי התאחדות המפה. וה-Local Planner, מתמודד עם פוקודות הניות, מתקן נתיבי ניוט קרובים ומפקד על Collision Avoidance (התחמקות ממכשולים). כרגע קיימים לרובוט Local Planner מסווג DWA שפירטנו עליו בחלק התיאורטי. יש לבנות Global Planner אשר מתאים למטרת הייעודי, כך שינויו בחלל תא הנוסעים בצורה מיטבית, ולסנן אותו עם ה-Local Planner.

על מנת ליעיל את הרובוט ולהביא אותו לאוטומטייזציה מלאה. נרצה שהרובוט יוכל להפעיל את זרועותיו באופן עצמאי ומלא. לשם כך יש לבנות רשת קובולוציה (CNN-Convolutional Neural Network) שתשזה את הסיטוטואצה הנוcharית וההתאמות לכך, תשלח פוקודות דרך הבקר לשאר חלקי הרובוט. למשל כאשר הרובוט נמצא צר (או צר במעטום בו יש קירות ולא כיסאות), הרשות זהה זאת, תעביר את הרובוט למצב Transition ורובוט יקפל את זרועותיו וינוע לנקודה הבאה בה הוא יוכל לפתחו אותן ולהמשיך את החיתוי.

כרגע לא קיים מקום רשמי למחשב הרובוט ולמצלמת הרובוט. יש ליצור מתקן שיכל להחזיק את שנירכיבי החומרה הללו באופן אידיאלי. המחשב צריך להיות רחוק מהמנועים שרוטטים ופולטים חום. והמצלמה צריכה להיות בגובה אופטימלי על מנת למפות בקרה מיטבית את החל תא הנוסעים של המטוס. הצעתי היא מתקן כפי שבנית בSIMOLIZER. ניתן לראות את המתקן באIOR 27.



מקורות

פרסומים באנגלית :

- [1] Guillaume Bresson, Zayed Alsayed, Li Yu, Sébastien Glaser. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, Institute of Electrical and Electronics Engineers, 2017, XX, pp.1. 10.1109/TIV.2017.2749181. hal-01615897.
- [2] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling".
- [3] Eric A. Wan and Rudolph van der Merwe, " The Unscented Kalman Filter for Nonlinear Estimation", Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006.
- [4] Dieter Fox, Wolfram Burgard, Sebastian Thrun, "The Dynamic Window Approach to Collision Avoidance", Dept. of Computer Science III, University of Bonn, D-53117 Bonn, Germany, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

קישורים למקורות באינטרנט :

- [5] Wikipedia – COVID-19 pandemic, https://en.wikipedia.org/wiki/COVID-19_pandemic
- [6] Wikipedia – Simultaneous localization and mapping, https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
- [7] YouTube – SLAM Course, Cyrill Stachniss, University of Freiburg, <https://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5QzbIHgl3b1JHimN>
- [8] Open SLAM – Gmapping, <https://openslam-org.github.io/gmapping.html>
- [9] First-Hand: The Unscented Transform, http://ethw.org/First-Hand:_The_Unscented_Transform
- [10] ROS – <http://wiki.ros.org>

דף נתוניים של רכיב :

- [11] Intel RealSense D435i (D400 series) – <https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf>
- [12] Nvidia Jetson Xavier NX – (1) <https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit>
(2) https://elinux.org/Jetson_Xavier_NX



נספחים

א. רשימת הקישורים של הפרויקט.

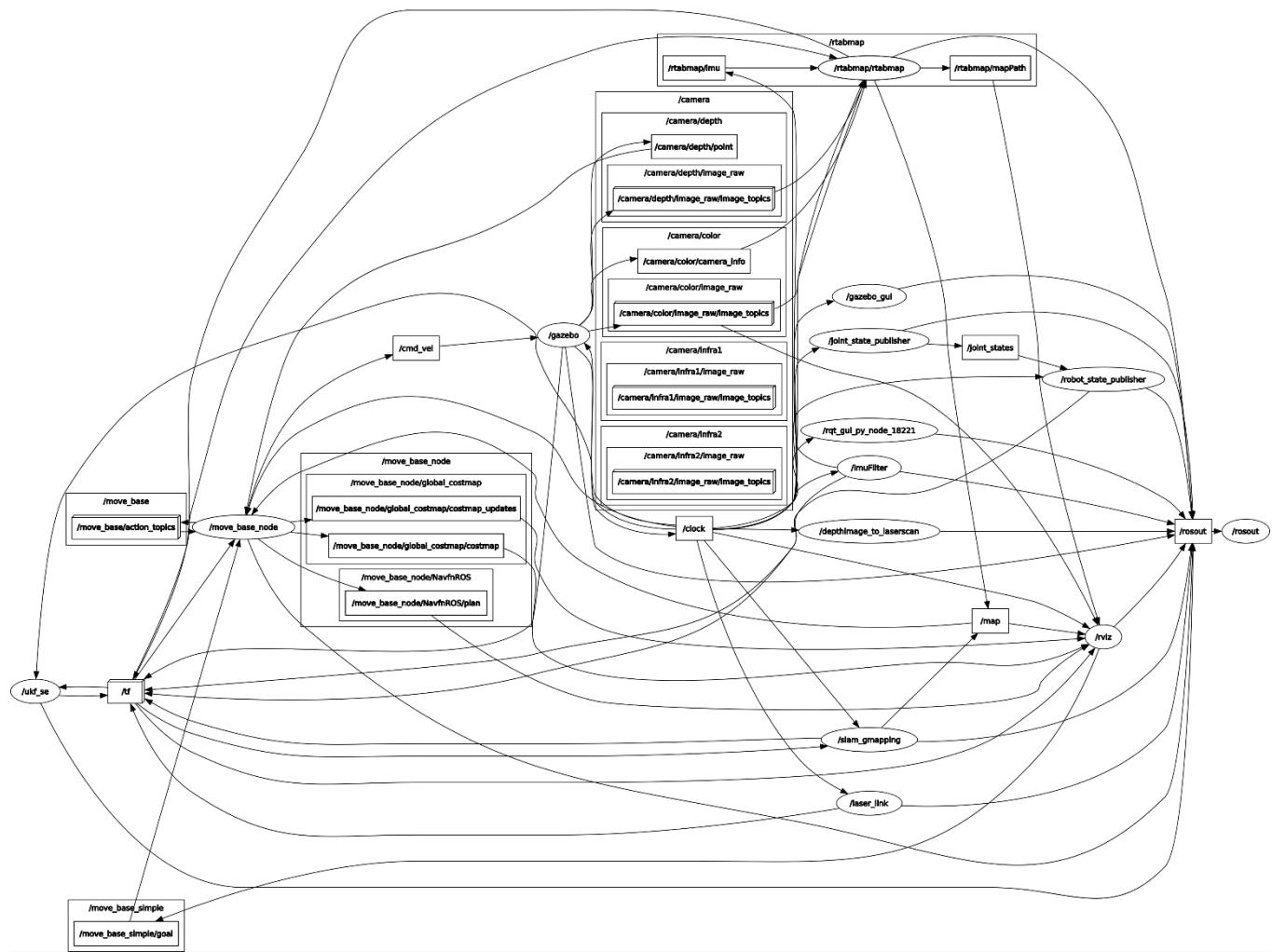
ערוץ ה-YouTube שלuproיקט:

<https://www.youtube.com/channel/UCwc1qD5aTrj4iaxQrpZu07w>

דף ה-GitHub שלuproיקט:

<https://github.com/Bengal1/AI-for-Robot>

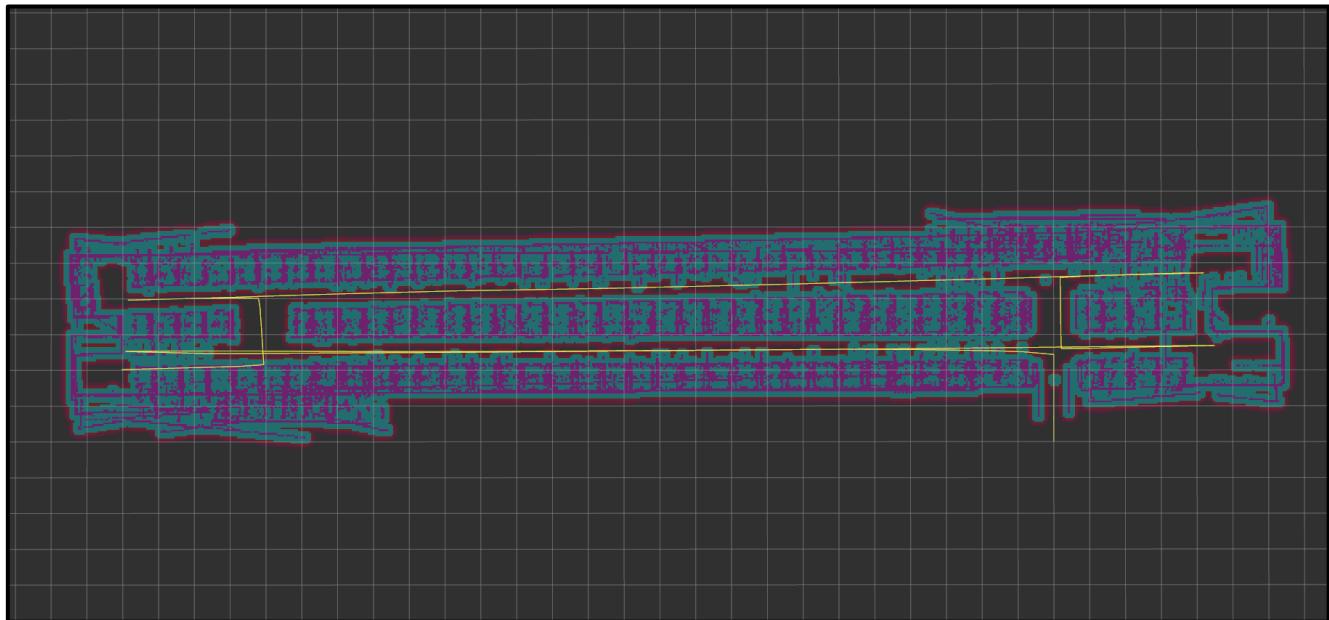
.Full RQT Graph .ב



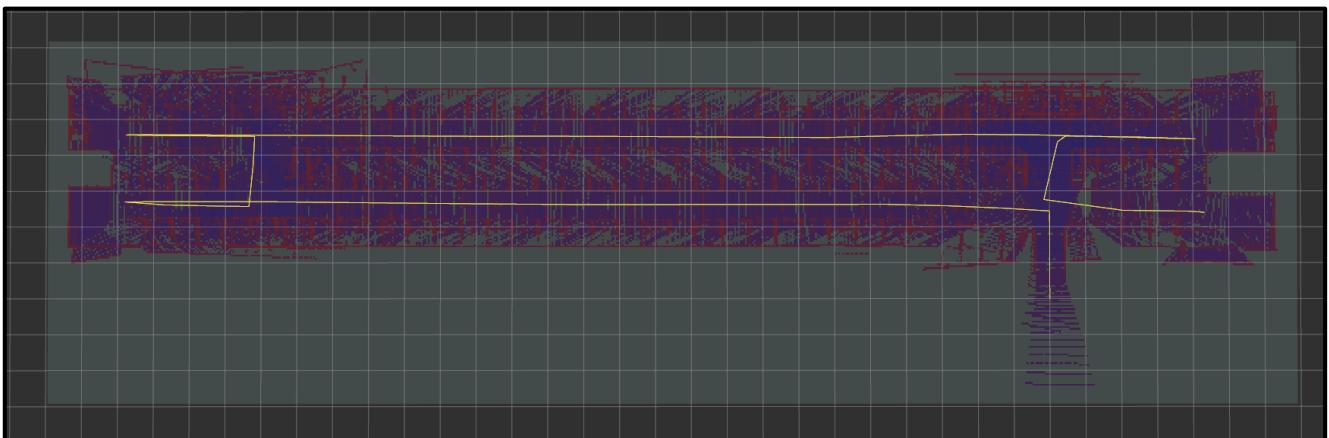


ג. מפות לשימוש פנימי של המערכת

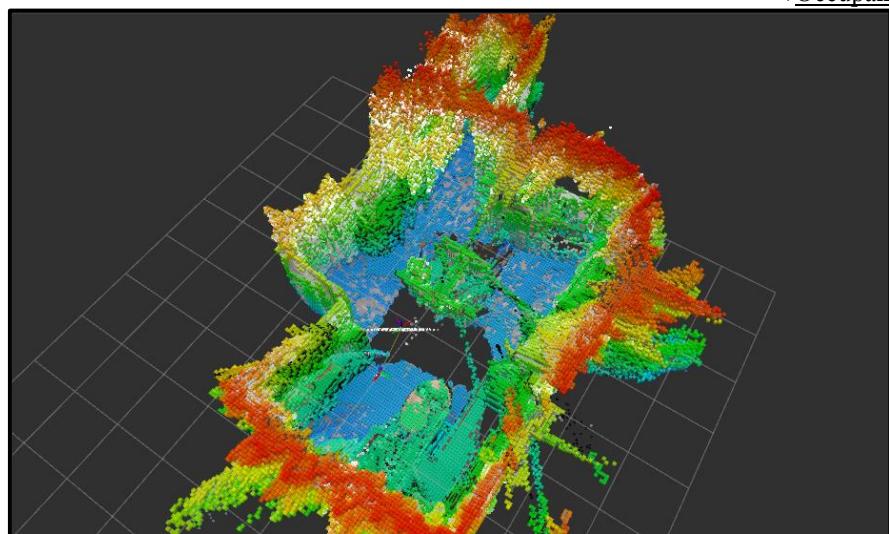
: Costmap



: RTAB-Map prob map



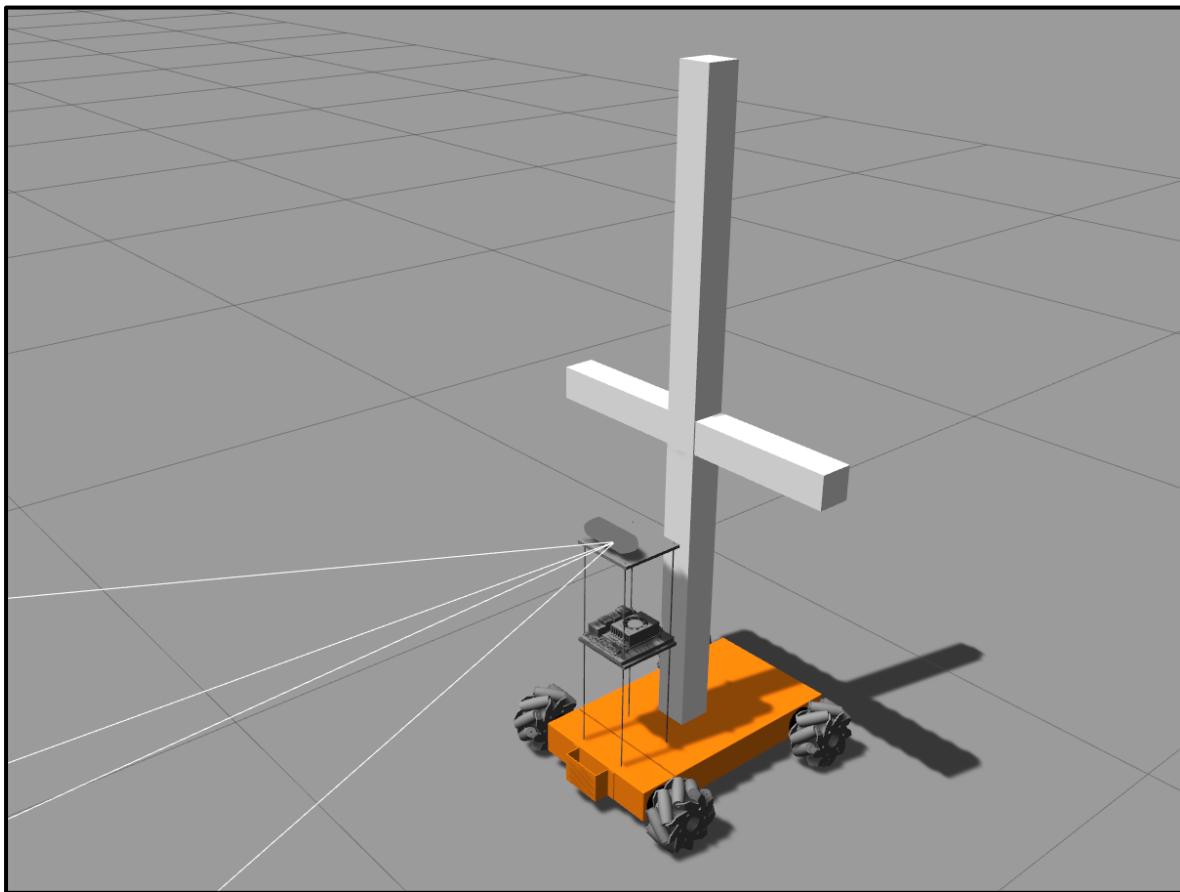
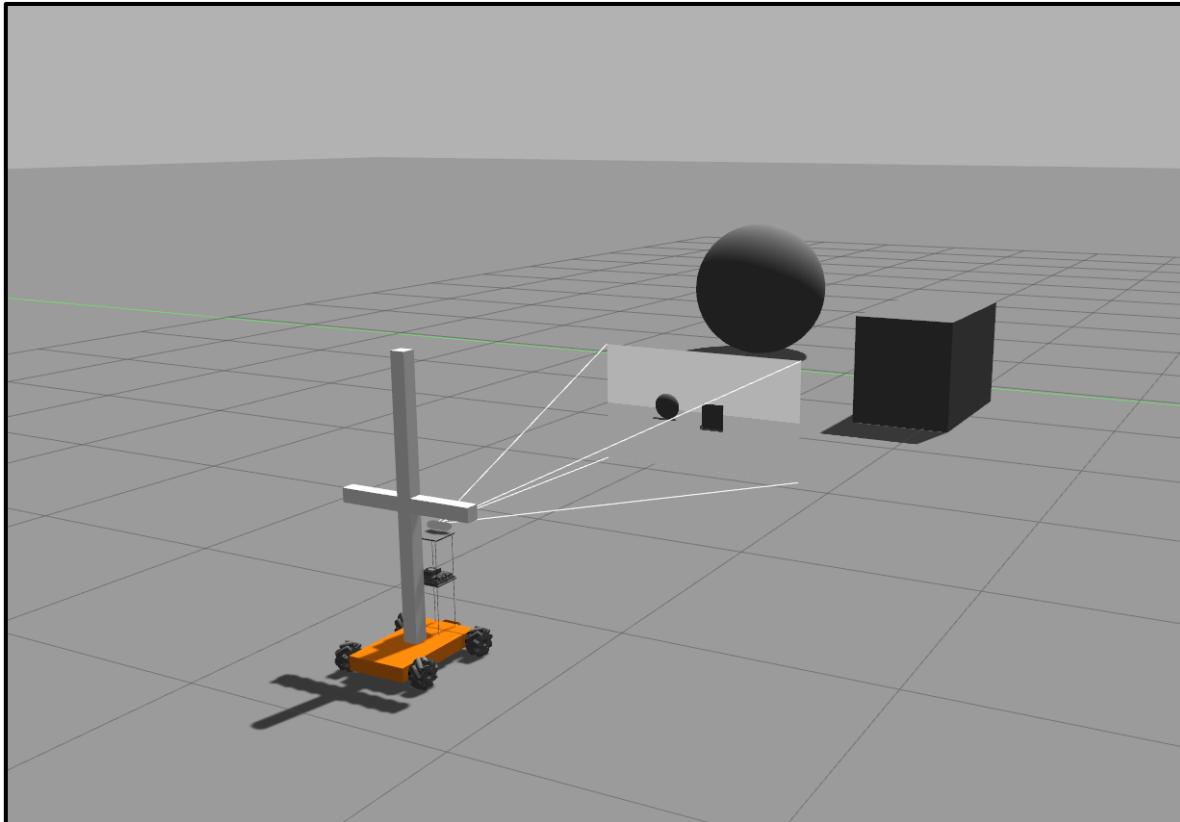
: Occupancy map

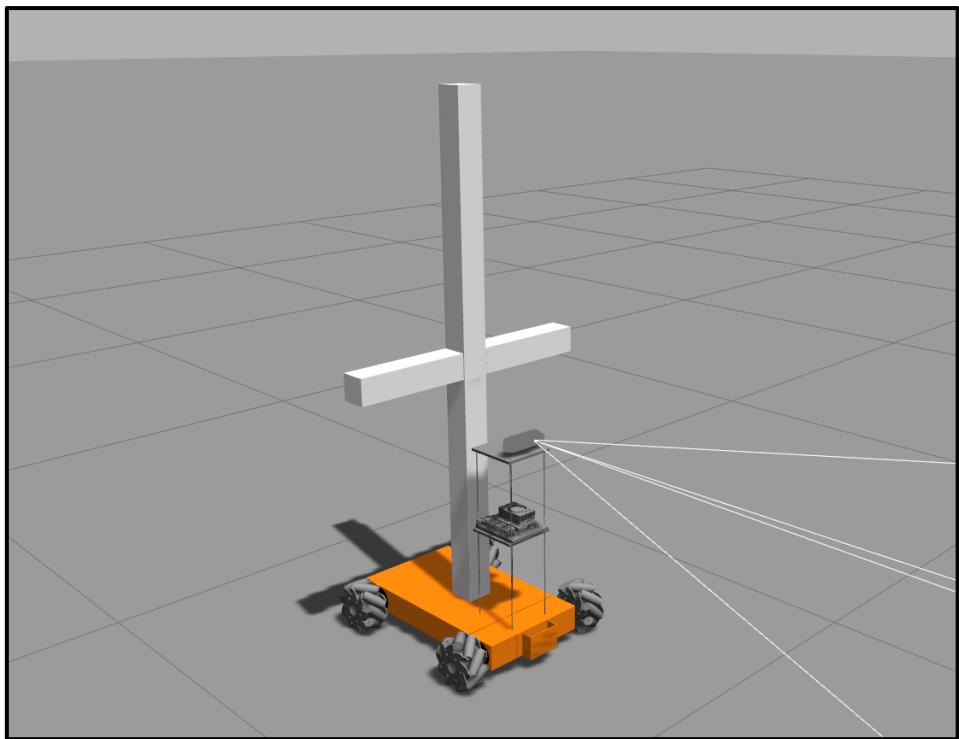




ד. גָּלוּחָה.

רַוְבוֹת הַסִּימּוֹלֶצִיה:



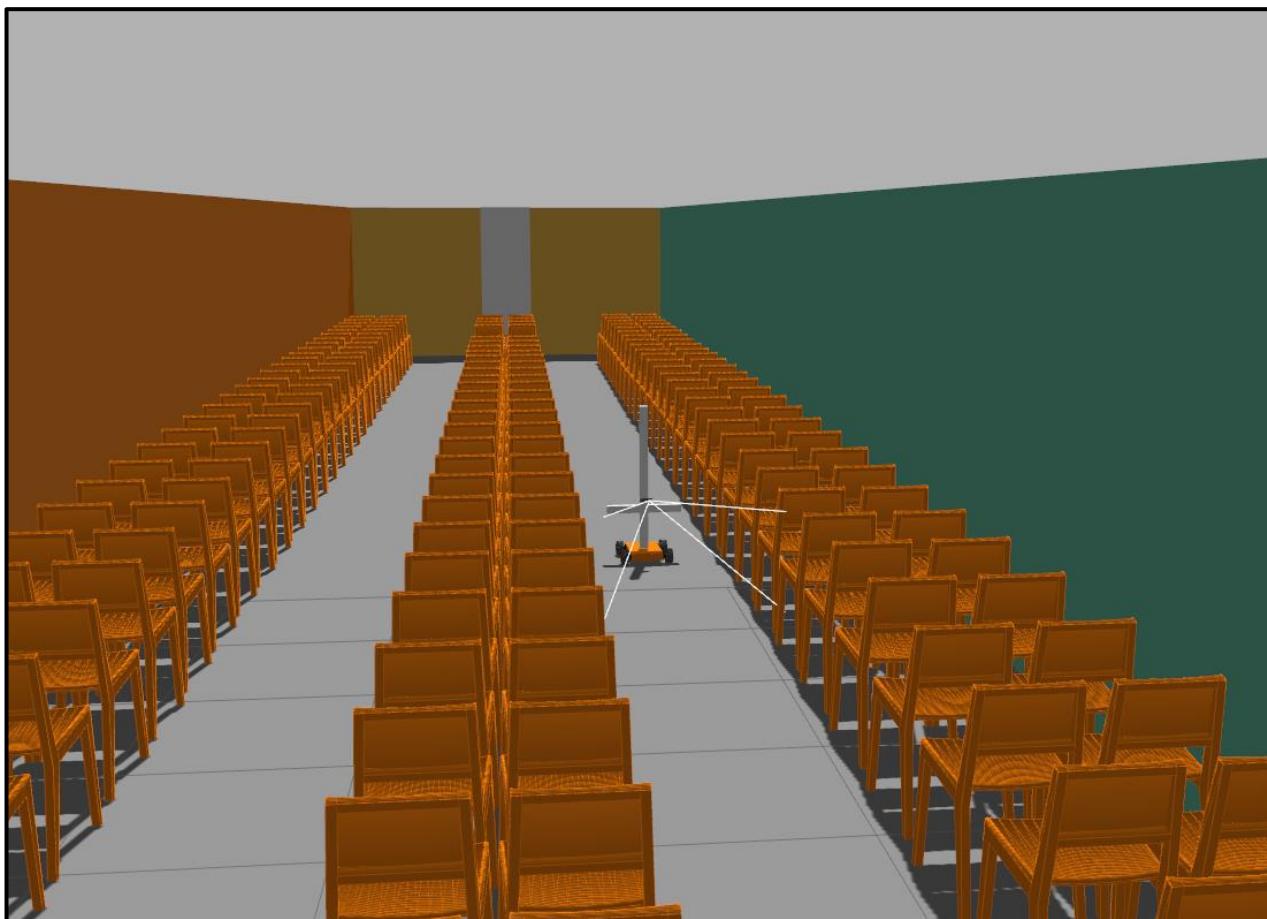
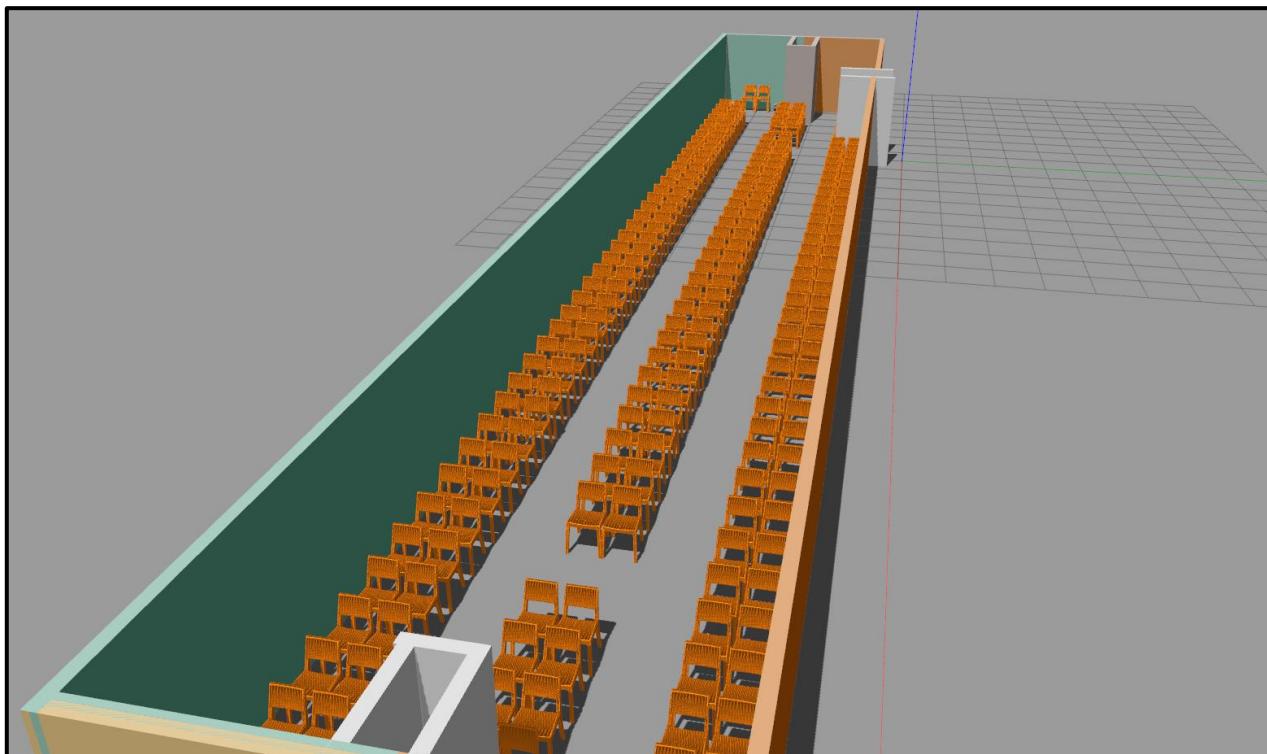


הרובוט האמיתי:





סביבת הסימולציה :





ערוץ ה-YouTube שלuproject

project_videos

No subscribers

HOME

VIDEOS

PLAYLISTS

CHANNELS

ABOUT

Uploads ▶ PLAY ALL

Video	Views	Published
flabmap slam results	3 views · 2 weeks ago	real flabmap slam
real slam result muted	4 views · 2 weeks ago	real slam result muted
Purebot simulation (speed X16)	7 views · 2 weeks ago	Purebot simulation (speed X16)
	5 views · 2 weeks ago	

CUSTOMIZE CHANNEL

MANAGE VIDEOS

Search

Home

Explore

Subscriptions

Library

History

Your videos

Watch later

SLAM Course (2013)

Show more

SUBSCRIPTIONS

TL-TV

Gordon Ramsay Hell...

Election Tube

CrashCourse

Preston Playz

Allegro Modafato

Padalic Edgington

Show 2 more

MORE FROM YOUTUBE

YouTube Premium



הוראות התקנה מדף GitHub של הפרויקט:

Installation Instructions

1. Install ROS - [ROS Installation Instructions](#)

2. Install ROS Navigation Package.

```
sudo apt-get install ros-melodic-navigation
```

For more information go to [ROS-Navigation](#)

3. Install Realsense Library for ROS.

- Installation of Intel realsense on Nvidia Jetson: [Jetson Installation Guide](#)

For more information go to [RealSense](#)

4. Install RTAB-Map:

```
sudo apt install ros-melodic-rtabmap-ros
```

For more information go to: [RTAB-Map-ROS](#)

5. Install ROS's Gmapping Package:

```
sudo apt-get install ros-melodic-gmapping
sudo apt-get install ros-melodic-navigation
sudo apt-get install python-opencv
sudo apt-get install python-numpy
sudo apt-get install python-scikits-learn
```

For more Information go to [SLAM Gmapping](#)

6. Install Depth image to Laser scan converter

7. Clone this repository:

```
cd ~/catkin_ws/src/
git clone https://github.com/Bengali/AI-for-Robot.git
cd ..
catkin_make
```

- (Optional) [RRT_Exploration Package](#).