-

ido zohar

**בקרת עכבה עם רובוט UR5**

בהנחיית ישראל שלהיים

לילך ביטון ; עידו זוהר

**תוכן עניינים**

[1) מבוא 2](#_Toc94388834)

[רקע תאורטי : 2](#_Toc94388835)

[השיטות: 2](#_Toc94388836)

[2) מטוטלת כפולה – בקרה על רובוט 2DOF: 4](#_Toc94388837)

[מידול המערכת: 4](#_Toc94388838)

[משוואות המצב של הרובוט: 4](#_Toc94388839)

[קינמטיקה ישירה: 4](#_Toc94388840)

[קינמטיקה הפוכה: 4](#_Toc94388841)

[מסלול: 5](#_Toc94388842)

[בקרה: 6](#_Toc94388843)

[הוספת כוח חיצוני: 6](#_Toc94388844)

[בקרת אימפדנס – Position Based 7](#_Toc94388845)

[תוצאות: 7](#_Toc94388846)

[3) בקרה על רובוט 6DOF בעזרת סימולציית MUJOCO 16](#_Toc94388847)

[סימולצית MuJoCo 16](#_Toc94388848)

[Peg In a Hole 16](#_Toc94388849)

[Sander 20](#_Toc94388850)

[מסקנות 27](#_Toc94388851)

# מבוא

## רקע תאורטי :

בקרת אימפדנס נועדה לסייע בביצוע משימות המערבות מגע עם הסביבה.

במאמר דנים ביתרונות וחסרונות של שתי שיטות מוכרות ומציעים שיטה חדשה המנסה לשמור על דיוק בעקיבה אחר האימפדנס ורובסטיות לאי ודאיות.

במסגרת הפרוייקט השתמשנו בשיטות:

1. Position Based – Impedance Control
2. Instantaneous Model - Impedance Control

### השיטות:

#### Dynamic Based Impedance Control

שיטה זו מתבססת על המודל הדינמי של המערכת. בהתאם חוק בקרה האימפדנס משתמש בעכבה הרצויה ובמודל הדינמי. שיטה זו אינה רגישה להפרעה, אך רגישה לטעויות במודל הדינמי אשר עשויות להוציא את המערכת מיציבות.

**Position Based Impedance Control**

שיטה זו מתבססת על מיקום המערכת. בהתאם חוק בקרה האימפדנס משתמש במיקום אשר נקבע על פי העכבה הרצויה ובמיקום האמיתי. השיטה מתבססת על תכנון מסלול, אותו משנים במידת הצורך על פי קבועי האימפדנס אותם בוחרים, כאשר קיים מגע עם הסביבה אשר משנה את המסלול המקורי.

משוואות המודל:

משוואת מודל האימפדנס היא:

מייצג את המיקום הרובוט במסלול המתוכנן המקורי.

מייצג את מיקום הרובוט החדש המתוכנן על פי מודל האימפדנס.

כאשר נממש את חוק הבקרה של הבקרים הפנמיים של הרובוט, נשתמש במיקום זה .

השיטה רובסטית לטעויות במודל הדינמי, אך רגישה להפרעה – ועשויה לצאת מיצבות.

**Instantaneous Model Impedance Control**

שיטה המשלבת בין היתרונת של השיטות הנ"ל.

השיטה משתמשת בתאוצה,מיקום ומהירות האמיתיים לצורך חיזוי המיקום של מודל האימפדנס הרצוי בצעד הזמן הבא.

המסלול הנדרש מחושב ע"י אינטגרציה של מודל האימפדנס בצעדי זמן, עם שימוש בתנאי ההתחלה (מיקום , מהירות) של התהליך האמיתי.

משוואות המודל:

משוואת מודל האימפדנס היא:

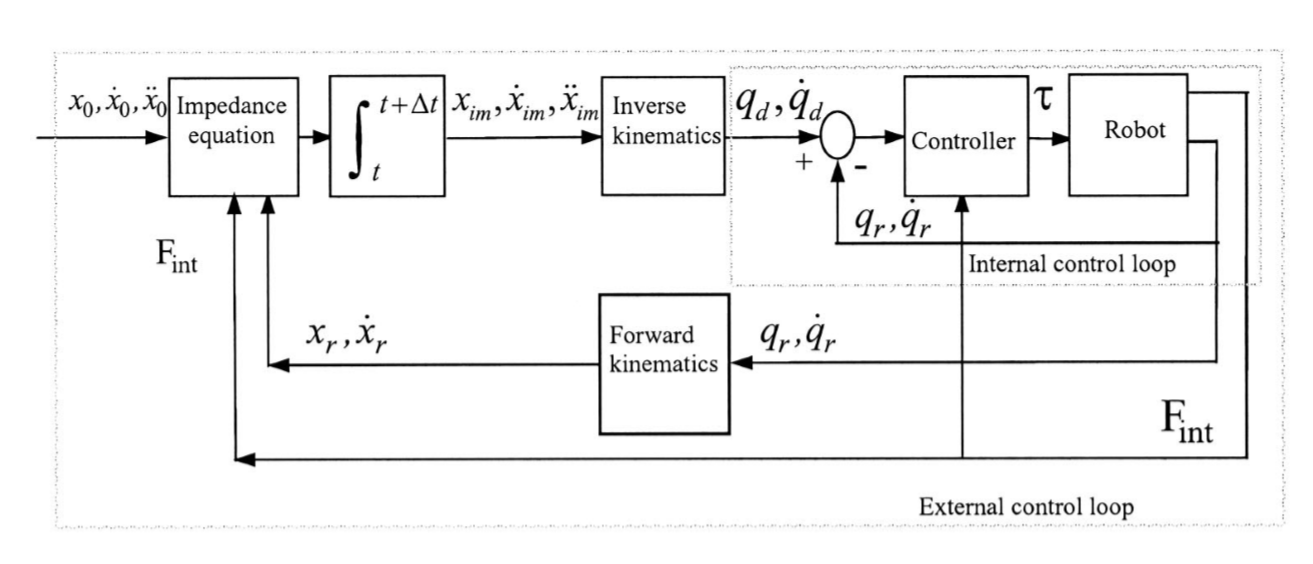
חישוב התאוצה של מודל האימפדנס הרגעי:

לקבלת מהירות ומיקום המודל מבצעים אינטגרציה על התאוצה בתוספת הקבועים המתאימים.

עבור מהירות מודל האימפדנס:

כאשר תנאי ההתחלה של מודל האימפדנס באינטגרציה הם נתוני התהליך האמיתיים. כלומר:

כאשר את המיקום האמיתי מוציאים מתוך הקינמטיקה הישירה של הרובוט, עם ערכי הזוויות הנוכחיים של המפרקים.



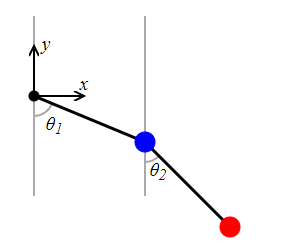
**דיאגרמת בלוקים עבור שיטת IM-IC:**

# מטוטלת כפולה – בקרה על רובוט 2DOF:

## מידול המערכת:

### משוואות המצב של הרובוט:

### קינמטיקה ישירה:

כאשר מערכות הצירים שלנו, וזוויות המפרקים מוגדרות באופן הבא, פיתחנו את הקינמטיקה הישירה של הרובוט:

### קינמטיקה הפוכה:

מידול הקינמטיקה ההפוכה הוא של רובוט בעל שתי דרגות חופש עם שני מפרקים סיבובים.

מצאנו מימוש [מתאים](https://robotacademy.net.au/lesson/inverse-kinematics-for-a-2-joint-robot-arm-using-geometry/). יש להמיר בסוף התהליך את הזוויות המתקבלות למערכת הצירים בה אנו עובדים. לצורך פישוט נסמן:

בהינתן מיקום יחידת הקצה:

המרה בין מערכות הצירים:

על מנת להמיר בין המהירות הקוויות למהירות הזוויתית נשתמש ביעקוביאן:

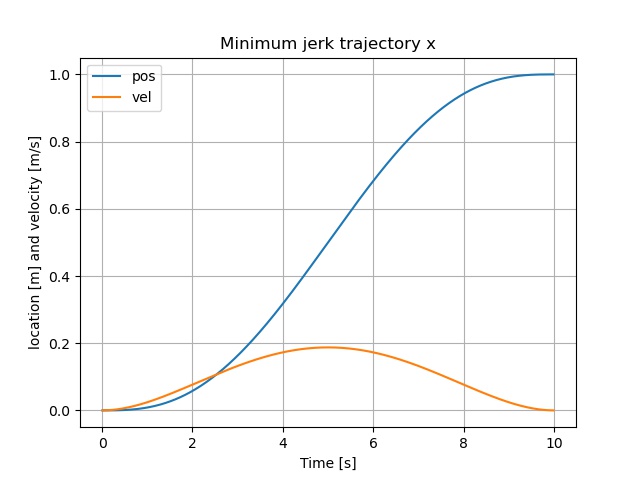
### מסלול:

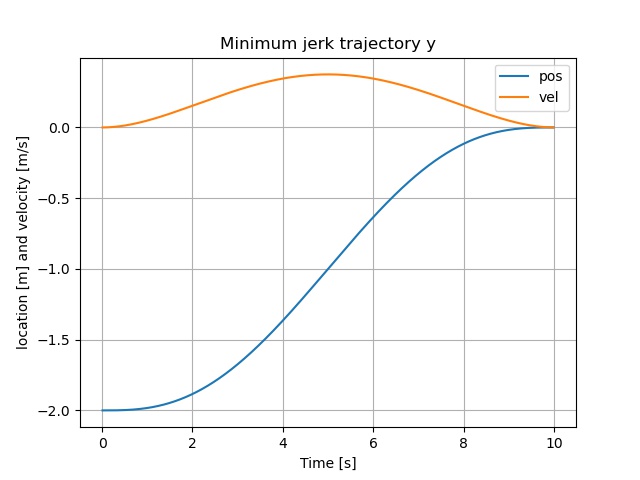
**בחרנו מסלול בעל פרופיל לינארי** minimum jerk trajectory**.**

**בהינתן נקודות התחלה וסיום, מחושבים ערכי המיקום**  **וערכי המהירות**  **בזמן, כך שמתקיים:**

**על מנת לקבל מסלול חלק ביותר** [**מצאנו**](https://mika-s.github.io/python/control-theory/trajectory-generation/2017/12/06/trajectory-generation-with-a-minimum-jerk-trajectory.html) **מסלול מיקום ע"י פולינום מסדר 5:**

**כאשר בחרנו שזמן התנועה הכולל יקיים מהירות ממוצעת של  *ולכן זמן התנועה המתקבל הינו .***

***ניתן לראות את המסלול שהתקבל בגרפים הבאים:***



**בעזרת הקינמטיקה ההפוכה והיעקוביאן נקבל את ערכי הוקטור המצב הרצוי אחריו נרצה לעקוב.**

### בקרה:

חוק בקרה PD עם פיצוי גרביטציה:

נממש את חוק הבקרה:

כאשר את קבועי הבקר בחרנו לפי ניסוי וטעיה, במטרה לצמצם את הרעידות ושגיאת המצב המתמיד – אך להישאר בתחום מאמץ בקרה נמוך במסגרת הפעולות שהרובוט מבצע.

בנוסף הוספנו אינטגרטור על מנת לאפס את שגיאת המצב המתמיד:

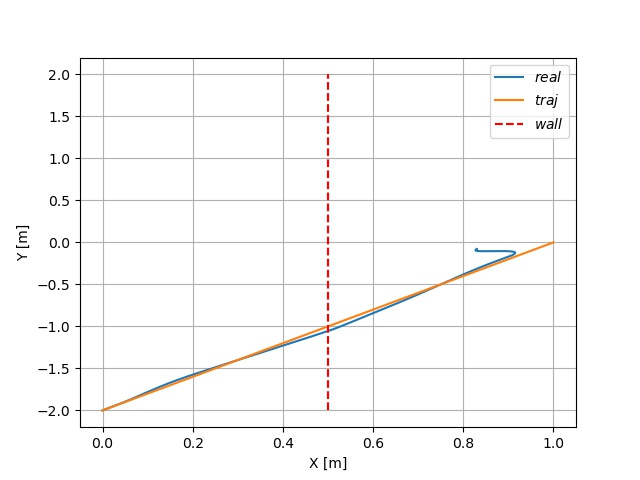
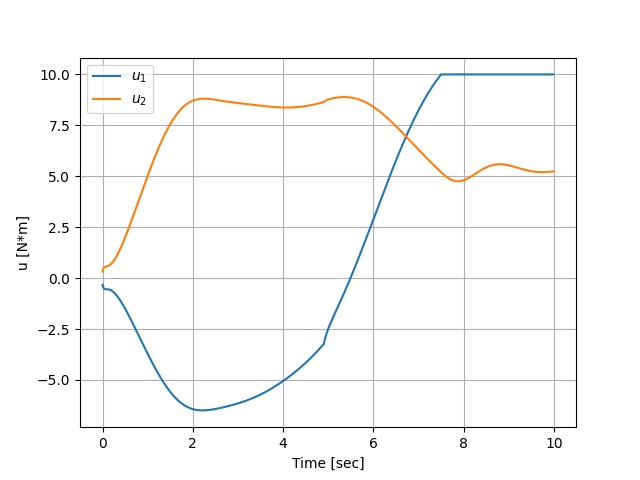
לבסוף הגדרנו כי מאמץ הבקרה המקסימאלי הוא

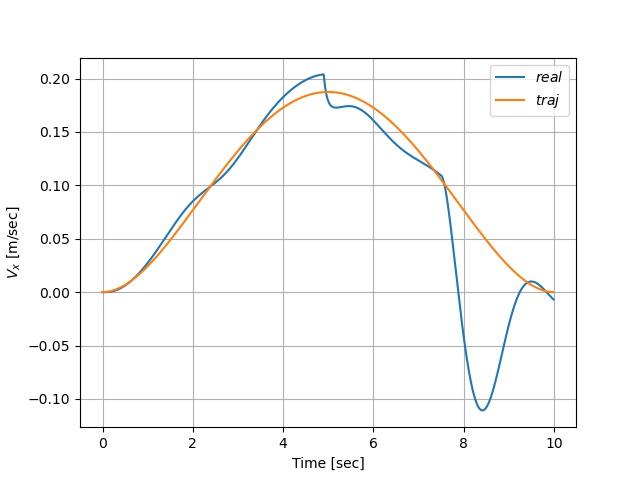
### הוספת כוח חיצוני:

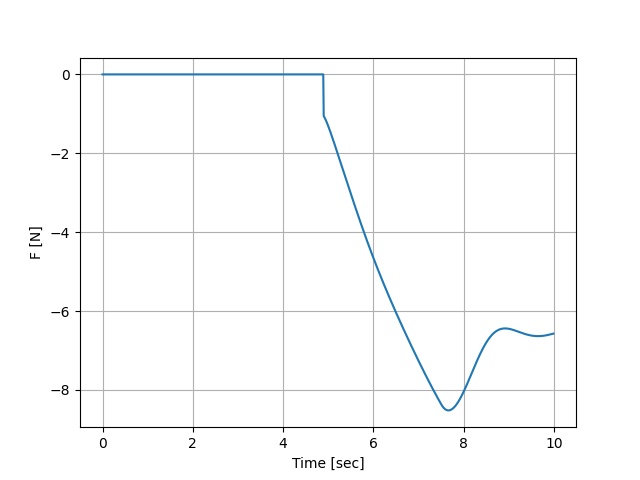
כוח המדמה 'קיר' המוגדר לפי המשוואה:

נרצה להמיר את הכוח הזה למרחב המפרקים על מנת להשתמש בו במשוואות התנועה:

***נבדוק כעת כיצד המודל מתנהג ללא בקרת אימפדנס ועם בקרת PID רגילה.***

***במקרה זה המסלול שהתקבל ומאמצי הבקרה הינם:***

***הכוח שיחידת הקצה הרגישה ומהירותה בעבור מקרה זה הינו:***

******

### בקרת אימפדנס – Position Based

**נוסיף חוק בקרה** Position-Based **שתוצאותיו יחליפו את**  **בוקטור המצב שמתקבל**

**כאשר נתאים את המערכת שתתנהג בהתאם לקבועי האימפדנס - תדירות העצמית הרצויה למערכת ומנת הריסון הרצויה:**

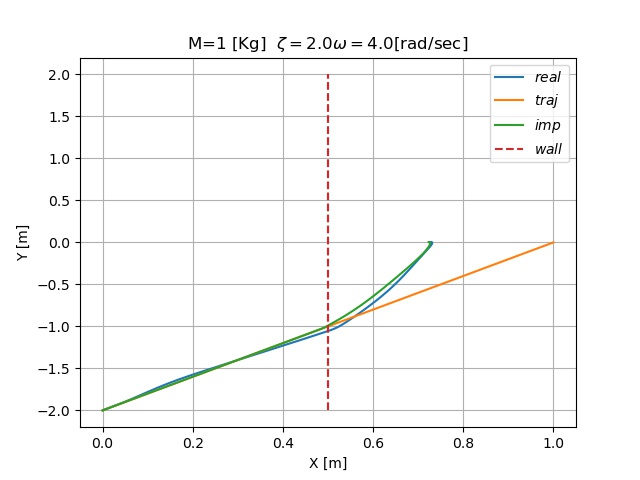
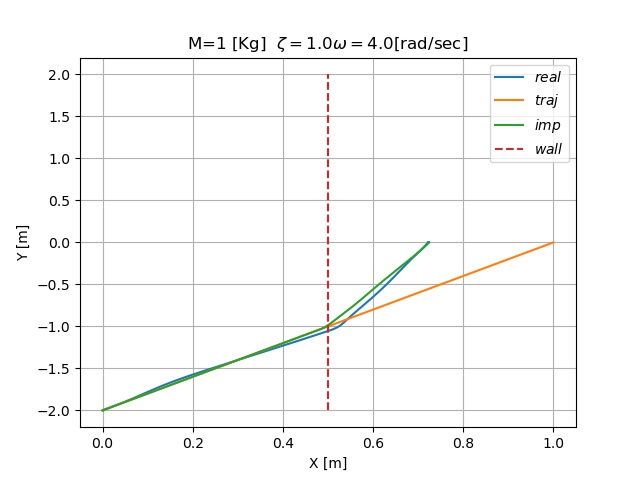
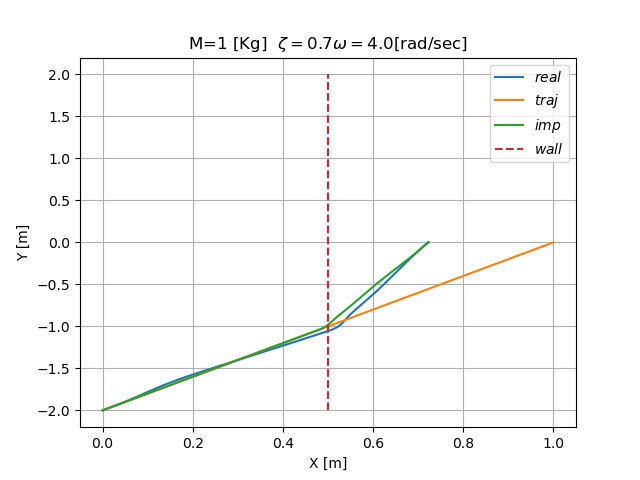
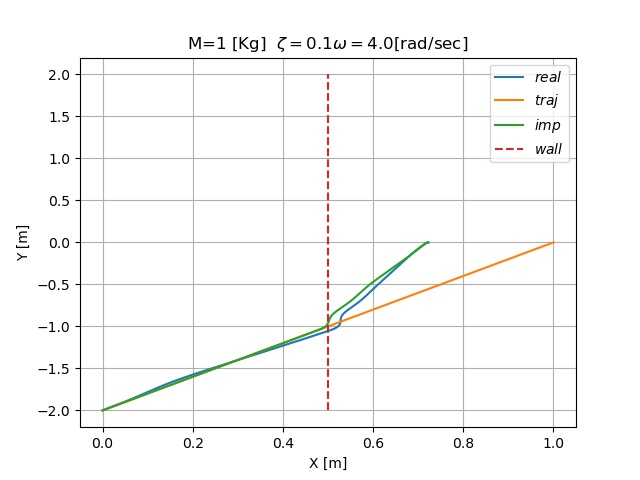
## תוצאות:

**נבחן את תוצאות הבקרה בעבור מספר קבועי אימפדנס שונים ונראה את ההבדלים ביניהם, כאשר הקיר ממוקם ב- וקשיחותו ובעל מקדם ריסון *.***

**כאשר דרשנו תנועה מנקודה אל הנקודה .**

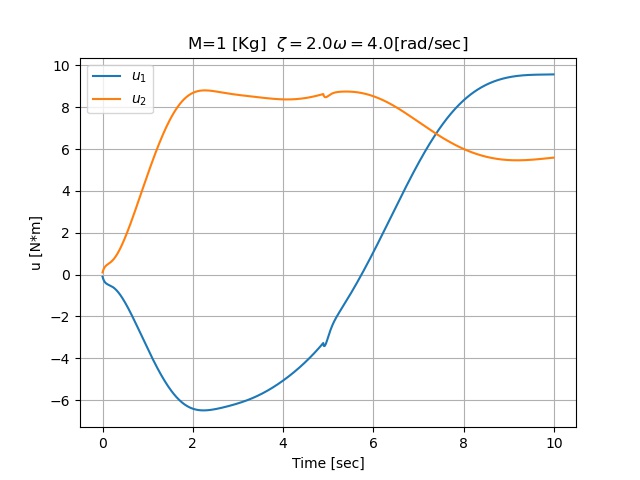
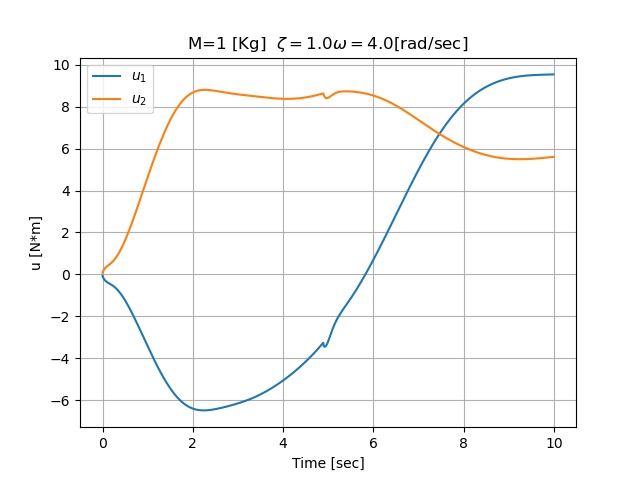
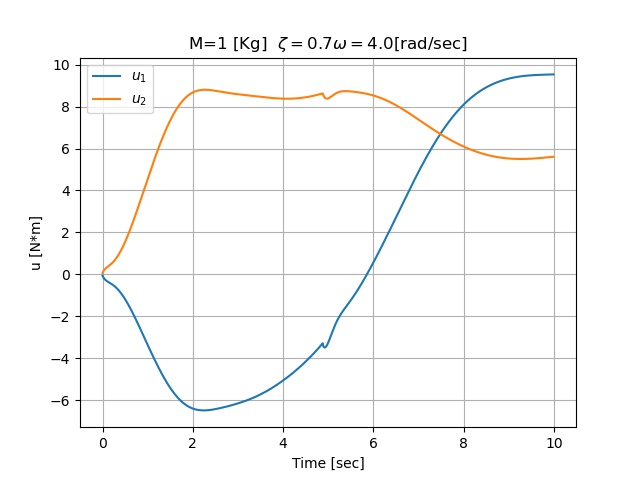
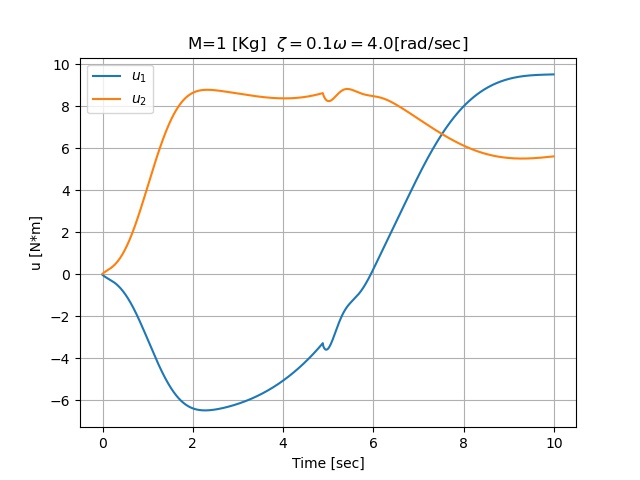
**נראה תחילה את ההשפעה של שינוי במנת הריסון של מודל האימפדנס.**

**בדקנו את התנהגות המערכת בעבור ערכי הריסון התקבל המסלול:**



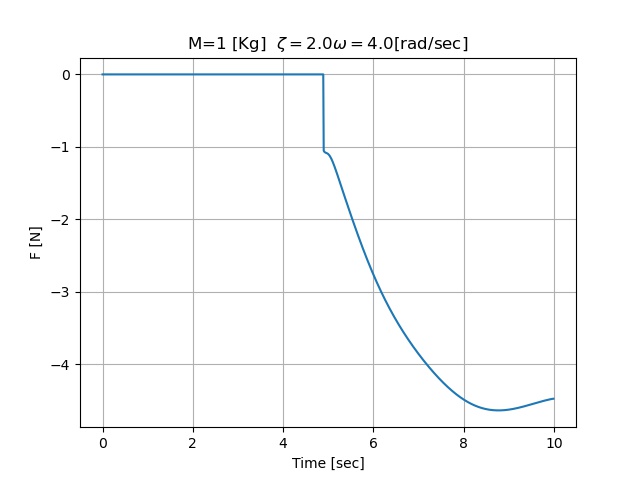
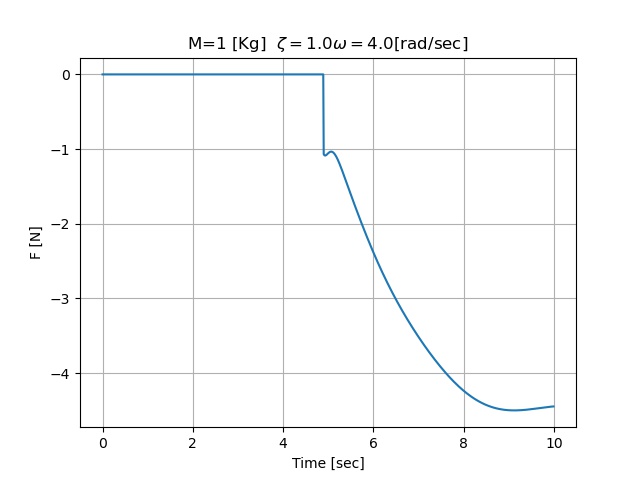
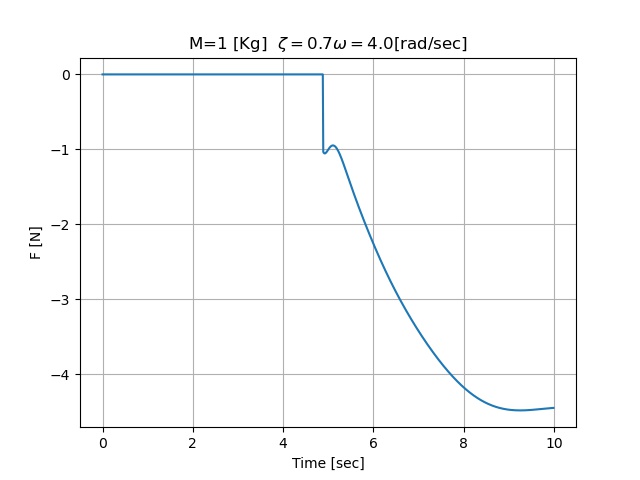
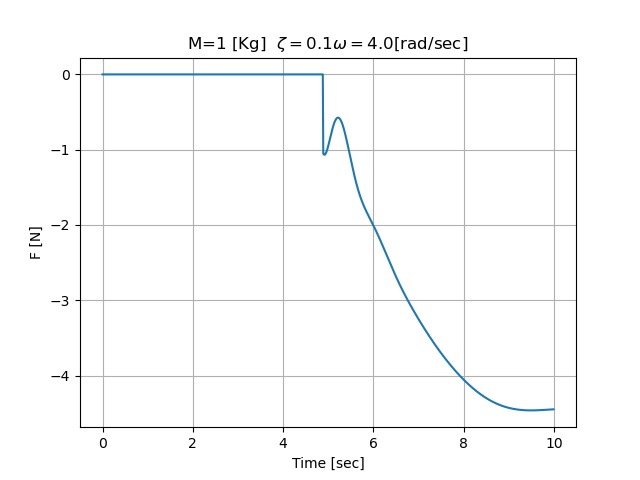
***ניתן לראות כי הריסון משפיע מעט על מסלול יחידת הקצה, ככל שהריסון גבוהה יותר, כך יחדת הקצה חווה פחות רעידות בזמן התנועה.***

***מאמצי הבקרה שהתקבלו בעבור כל אחד מהמקרים הינם:***



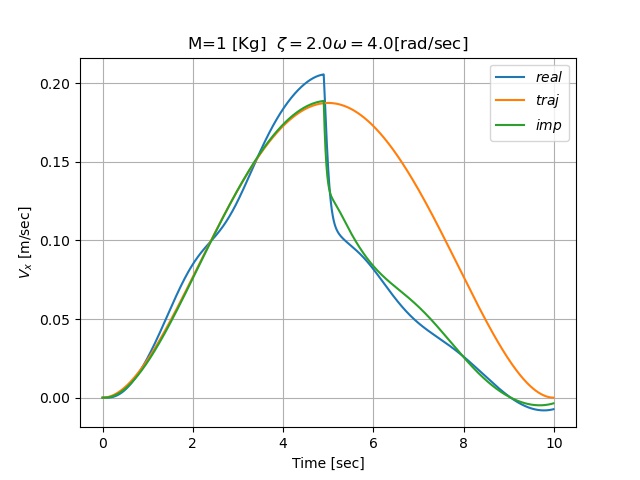
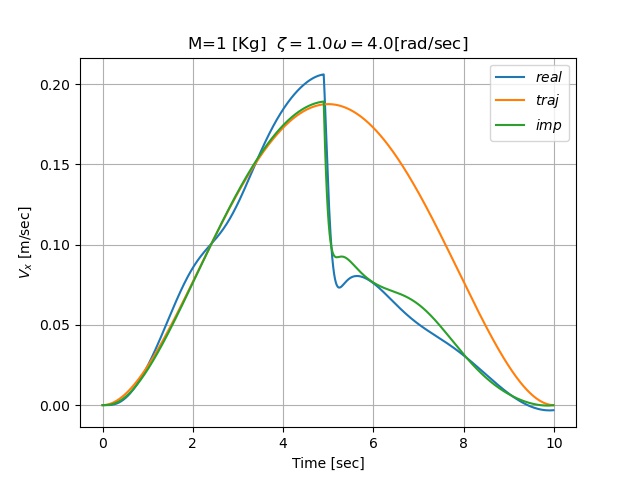
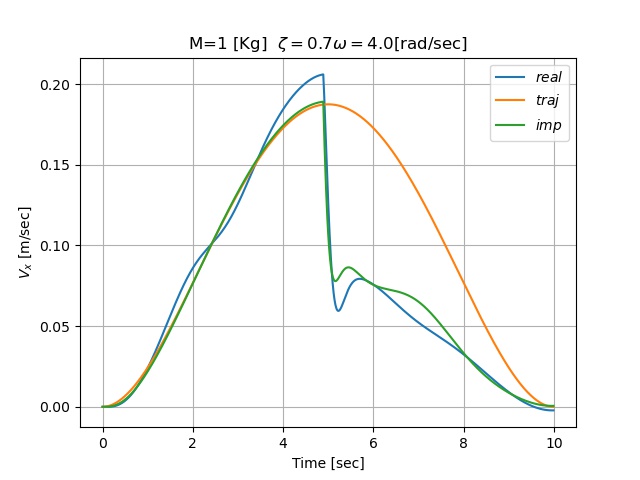
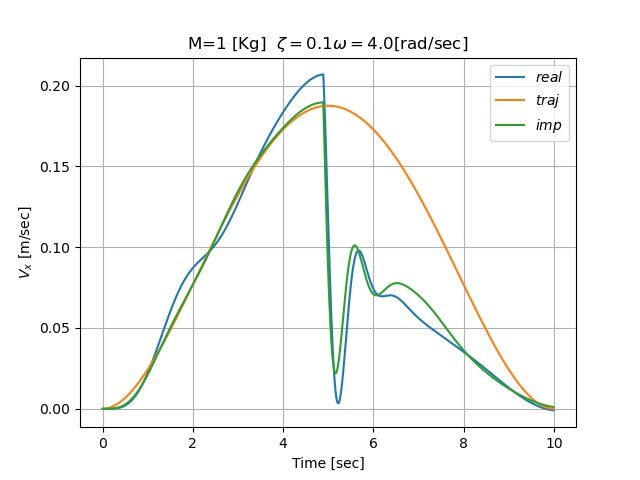
***גם על מאמץ הבקרה אין השפעה ניכרת. ניתן לראות שמאמץ הבקרה קטן – ברגע הפעלת הכוח – ככל שהריסון גדל, אם זאת אין לריסון השפעה ניכרת מעבר לכך ומאמץ הבקרה נשאר באותו תווך ערכים.***

***הכוח שיחידת הקצה הרגישה בעבור כל אחד מהמקרים הינו:***



***בגרף הכוחות ניתן לראות כי הריסון כמעט ולא משפיע על הכוח המקסימלי אותו חווה יחידת הקצה, אך הוא משפיע על האופן בה התפסנית מגיבה כאשר היא נתקלת בהפרעה.***

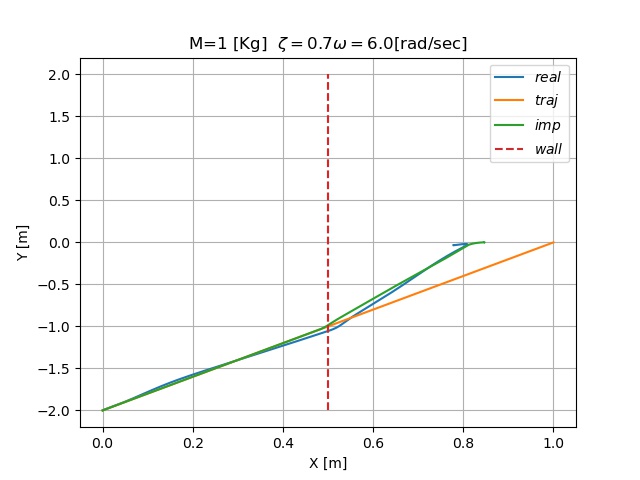
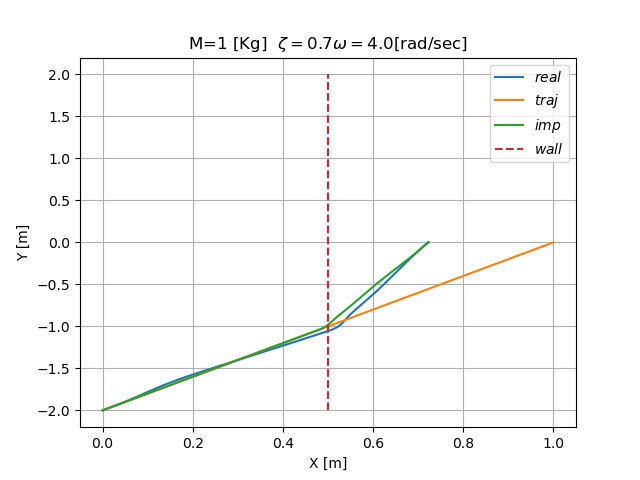
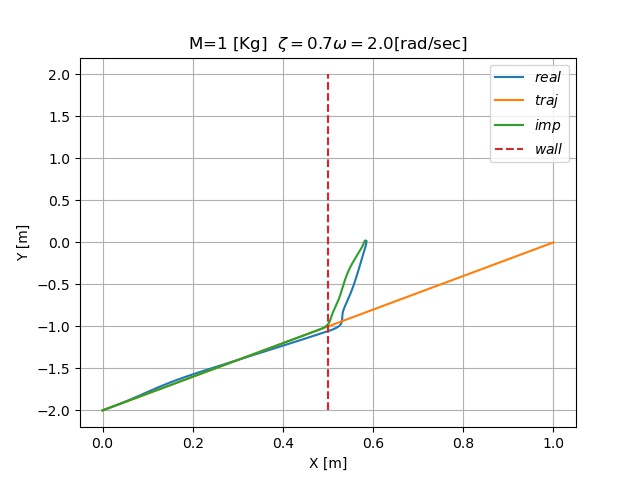
**מהירות *יחידת הקצה בעבור כל אחד מהמקרים הינו:***



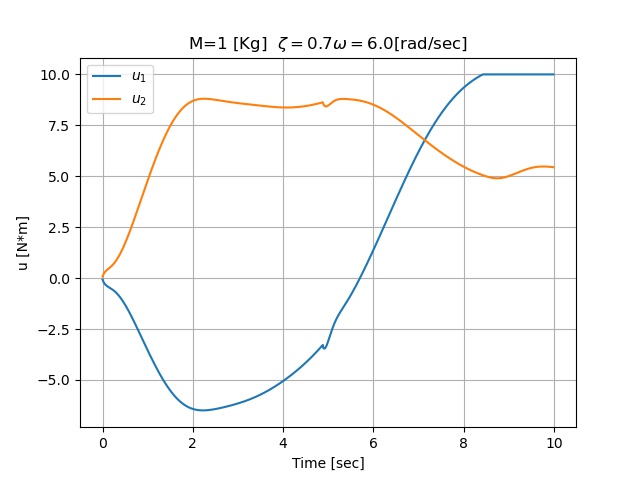
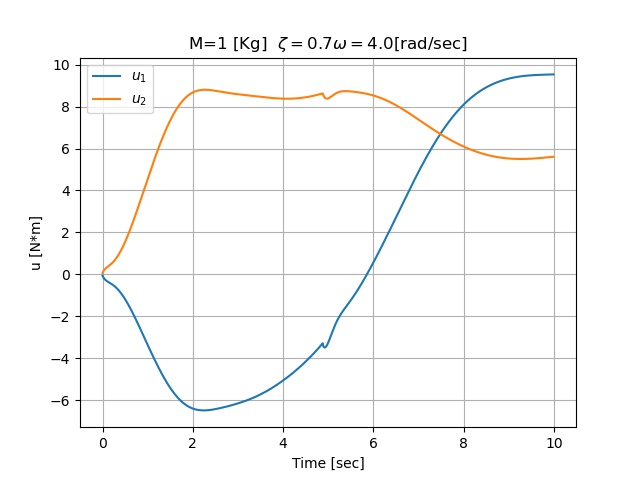
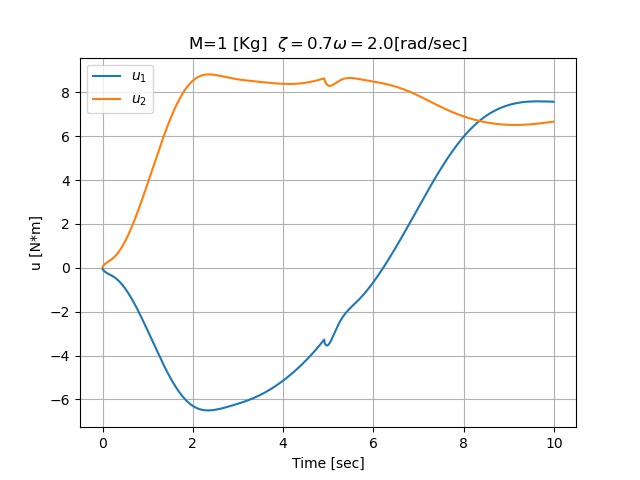
***בגרף המהירות ניתן לראות כי ריסון חזק ממתן את השינוי במהירות, בעוד ריסון נמוך יגרום לשינויים חדים במהירות במגע עם ההפרעה.***

***לסיכום, ערך ריסון גבוהה יותר משפר את התגובות של המערכת כאשר היא נתקלת בהפרעה, אך אין לו השפעות ניכרות על התוצאה הסופית.***

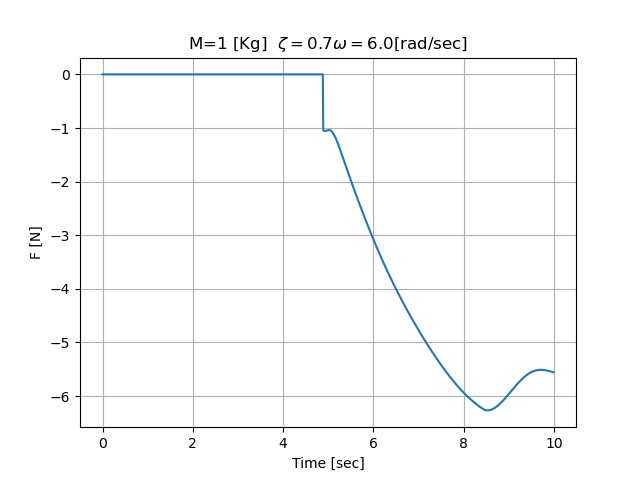
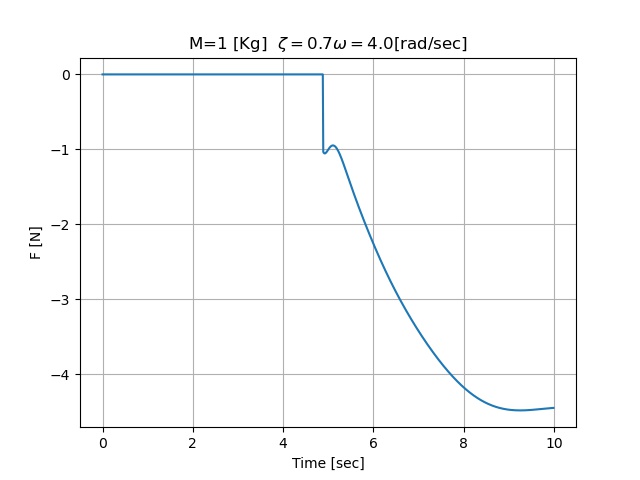
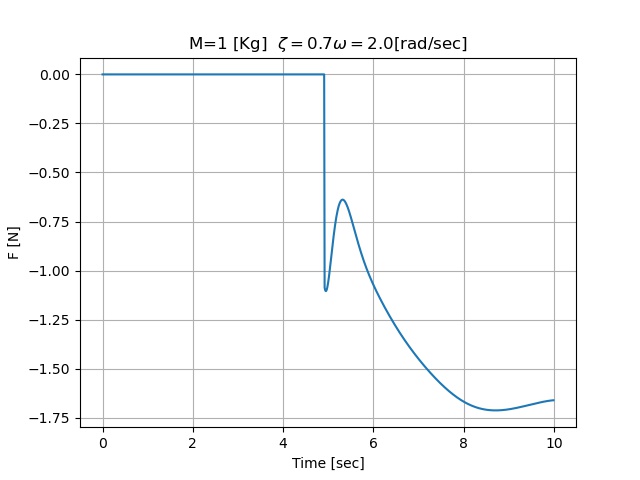
**בדקנו את התנהגות המערכת בעבור התדירויות העצמיות התקבל המסלול:**



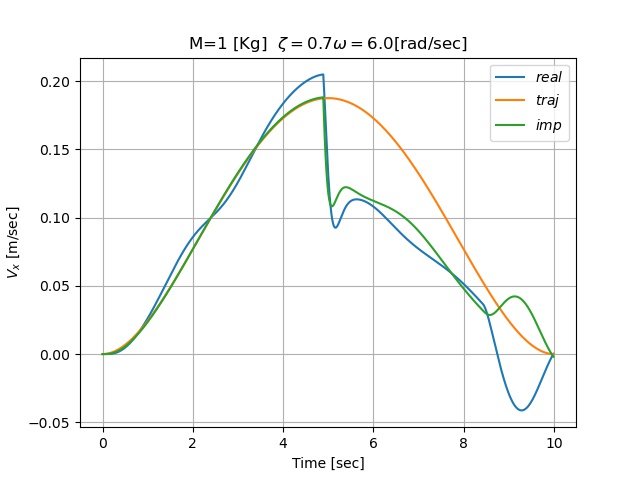
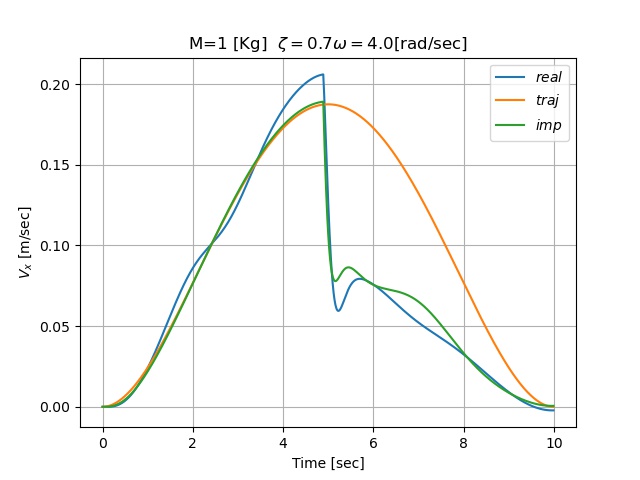
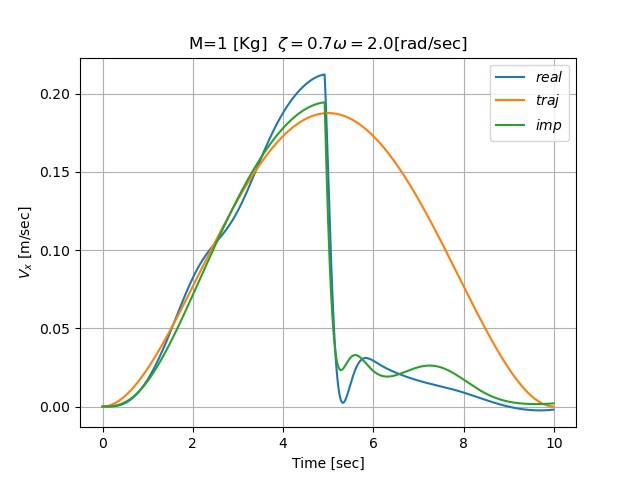
***מאמצי הבקרה שהתקבלו בעבור כל אחד מהמקרים הינם:***



***הכוח שיחידת הקצה הרגישה בעבור כל אחד מהמקרים הינו:***



**מהירות *יחידת הקצה בעבור כל אחד מהמקרים הינו:***



***שינוי בתדירות העצמית גורר שינויים משמעותיים בהתנהגות המערכת.***

***בחירת תדירות עצמית גבוהה יותר תניב עקיבה טובה יותר אחרי המסלול הרצוי ושינויים איטיים יותר במהירות יחידת הקצה – בפגישה בהפרעה, אך מאמצי הבקרה והכוח הפועל על יחידת הקצה יגדול בהתאם ואופן משמעותי.***

***בנוסף ,שימוש בחוק בקרת אימפדנס – כתלות בערכי התדירות העצמית שבחרנו – מאפשר לנו להגיע לתוצאות כמעט זהות למימוש בבקר PID ללא בקרת אימפדנס ולהוריד את הכוח המקסימלי אותה חווה יחידת הקצה ואת מאמצי הבקרה.***

# בקרה על רובוט 6DOF בעזרת סימולציית MUJOCO

### סימולצית MuJoCo

MuJoCo הוא מנוע פיזיקלי המאפשר לדמות, באופן וירטואלי, סביבת עבודה של רובוטים מורכבים, ולחקור את ההתנהגות שלהם בסביבות שונות בביצוע פקודות שונות.

בפרוייקט השתמשנו ברובוט UR5 של חברת Universal Robots, רובוט בעל 6 דרגות חופש.

המנוע מקבל תיאור של הרובוט, בעזרתו הוא בונה את הסביבה הוירטואלית ומחשב את הפיזיקה של הרובוט (קינמטיקה ישירה, יעקוביאן).

מתוך הסימולציה ניתן להוציא את ערכי המנועים של הרובוט – זווית, מהירות, מומנטים.

## Peg In a Hole

**מבוא:**

Peg In a Hole הוא משחק בו המטרה היא להתאים בין צורה לבין החור המתאים לה, וכך להכניס את האובייקט לתוך קופסה. בפרוייקט קודם שנעשה, האתגר היה להכניס גליל לתוך חור עגול – כך שאין חשיבות לאוריינטציה של הגליל כל עוד הוא מאונך למישור הקופסה. בפרוייקט זה החלפנו את הגליל בקוביה, על מנת לחקור את מודל האימפדנס בסביבה מורכבת יותר. הכנסנו לרובוט הפרעה – הכרחנו אותו להתנגש בקופסה (הכנסנו שגיאה קטנה של מיקום החור) וניסנו לגרום לו להיכנס לתוך החור המתאים – כאשר את המיקום הסופי הוא יודע.

על מנת לחקור את פעולות הרובוט באופן מציאותי יותר, הוספנו חיישן כוח (צבוע באדום) במיקום בו הוא נמצא על הרובוט במציאות. החיישן ממוקם בין יחידת הקצה לבין החוליה הקודמת לה.

בנוסף, ביצענו נירמול של הכוחות על מנת שהחיישן יתעלם ממשקל יחידת הקצה ונוכל להתייחס לכוחות אותם אנו חווים בלבד.

A picture containing floor, toy

Description automatically generated

**חוק בקרה:**

בסימולציה זו ביצענו בקרה היברידית המשלבת בין בקרת PID לבין חוק הבקרה Instantaneous Model Impedance Control , כאשר חוק הבקרה הינו PID בלבד עד רגע פגיעת החלק במשטח המשחק (אשר מזוהה באמצעות סנסור הכוח), שם חוק הבקרה מתחלף להיות Instantaneous Model Impedance Control כפי שמתואר במבוא. כך שאם המסלול ומתאר השטח ידוע במדויק ומאפשר הכנסת החלק ללא שגיאה וללא פגיעה במשטח המשחק, חוק הבקרה ישאר PID, ואילו רק במקרה של פגיעה בחלק המשטח, כתוצאה משגיאה או כתוצאה מאפיצות קרובה, חוק הבקרה יתחלף לבקרת אימפדנס המותאמת למקרים אלו.

חוק בקרה Instantaneous Model Impedance Control:

משוואות חוק הבקרה הינן מבוססות על אלו המתוארות במבוא, כאשר על מנת לתת לרובוט יכולת לסובב את החלק, ביצענו צימוד – של איברים ספציפיים – במטריצת הקשיחות של האימפדנס.

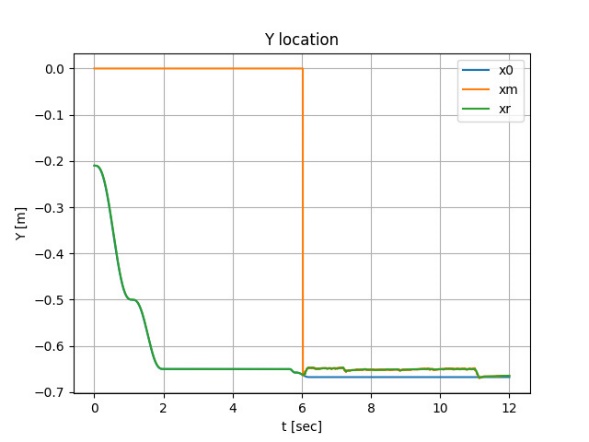
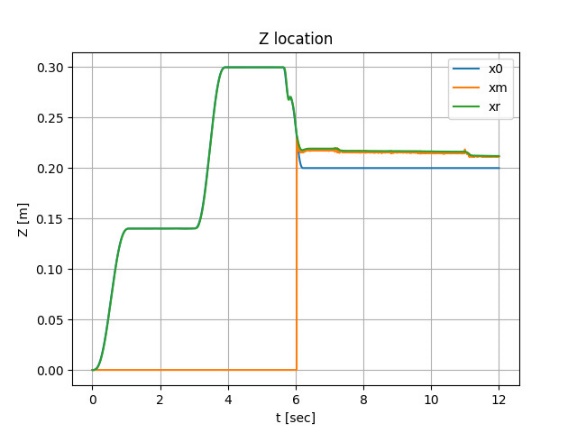
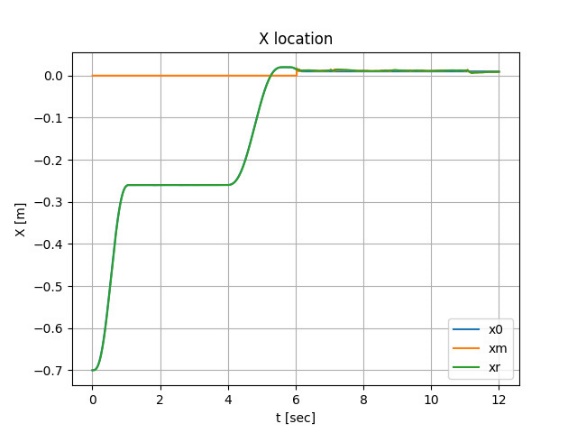
מטריצת האימפדנס:

**תוצאות:**

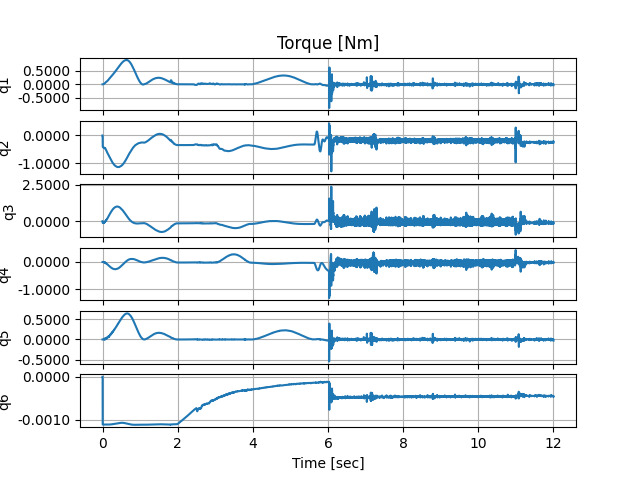
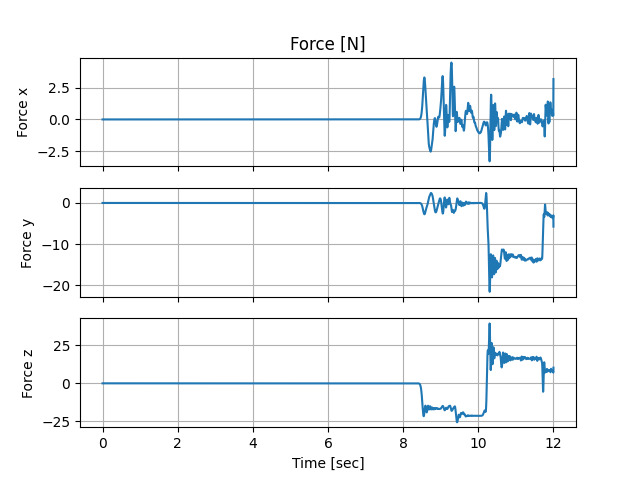
השגיאה במיקום הסופי: :

הרובוט הצליח להכניס את הקוביה לחור!

מסלול יחידת הקצה:



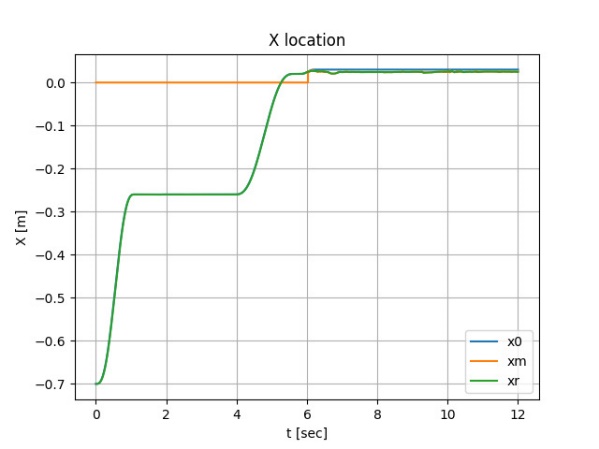
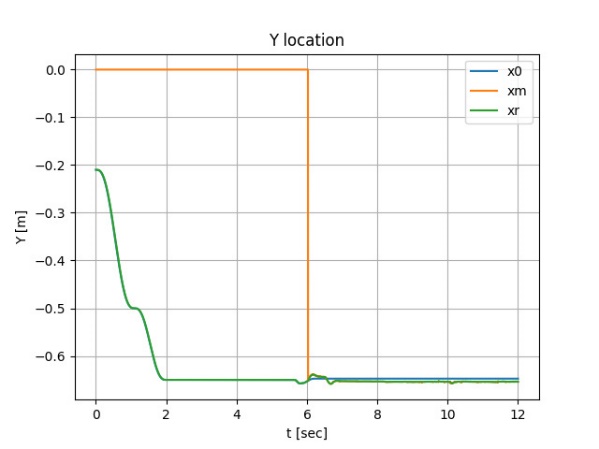
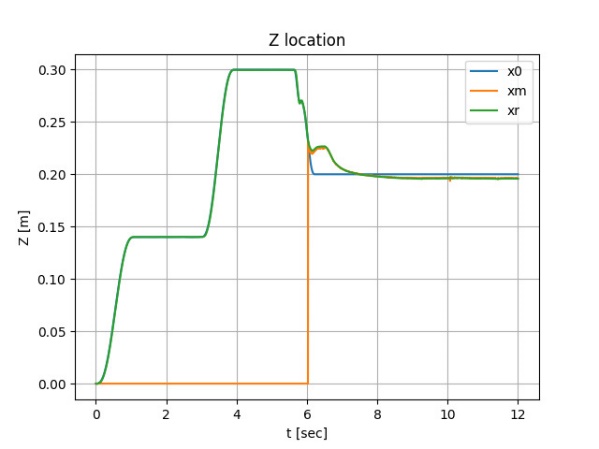
*הכוחות בחיישן והמומנטים במפרקים:*



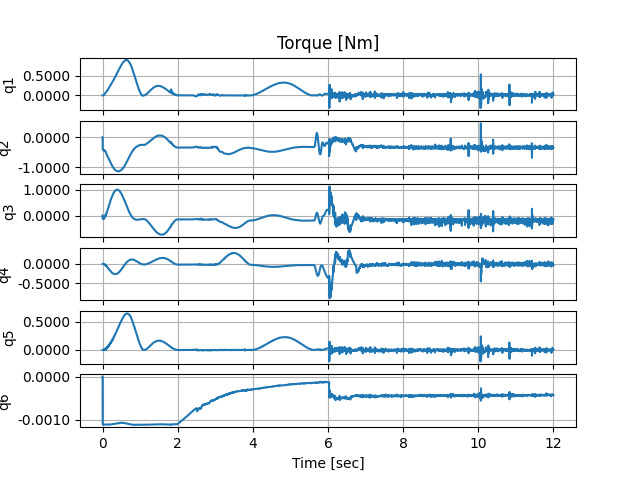
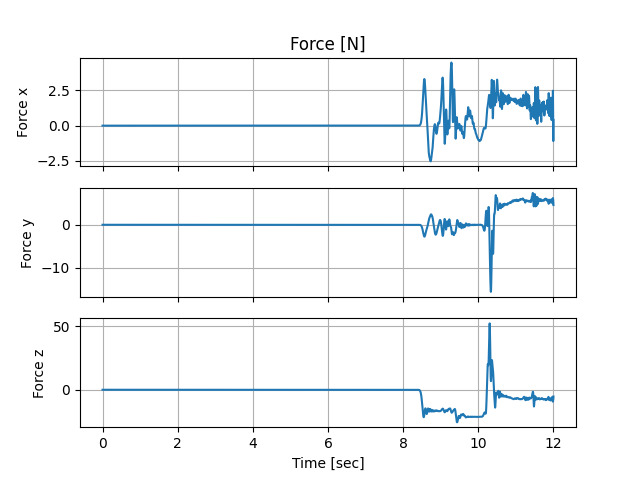
השגיאה במיקום הסופי: :

הרובוט הצליח להכניס את הקוביה לחור!

מסלול יחידת הקצה:

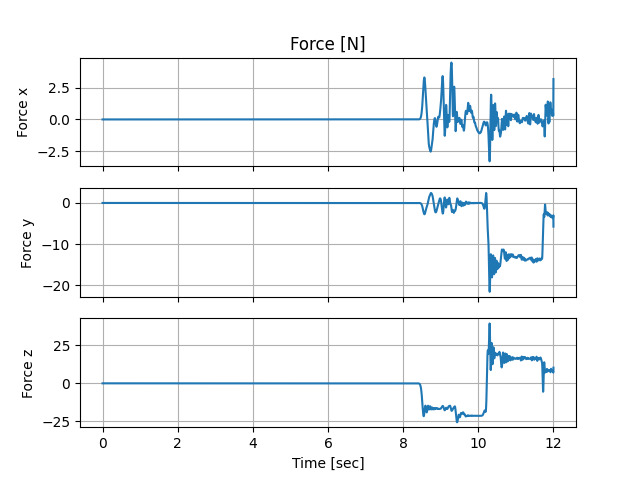
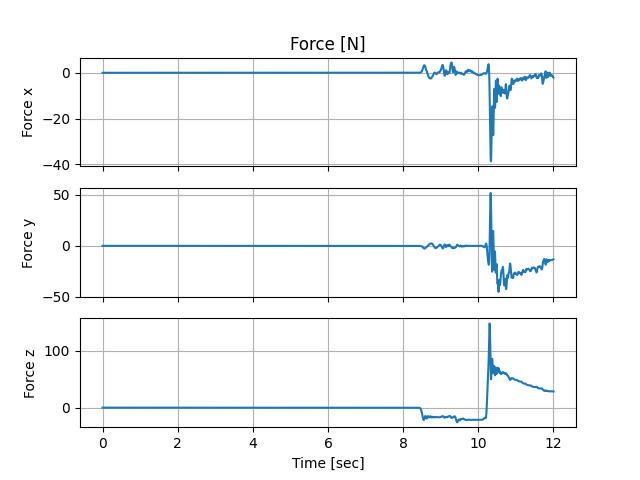


*הכוחות בחיישן והמומנטים במפרקים:*



בנוסף, נשווה בין התוצאות אשר מספקת בקרת העכבה, לבין התוצאות שחוג בקרה פשוט מסוג PID נותן.

עבור שגיאה במיקום הסופי :



בקרת עכבה

בקרת PID

ניתן לראות כי בקרת העכבה משיגה את המטרה שלה ומקטינה את הראקציות בצורה משמעותית.

## Sander

**מבוא:**

בחלק זה של הפרוייקט, נתנו לרובוט משימה חדשה - לשייף משטח לאורך זמן וסביב כוח קבוע. לצורך פעולות אלו, החלפנו את יחידת הקצה מתפסנית לראש משייף. בנוסף הוספנו קיר מול הרובוט אשר ישמש כמשטח אותו הרובוט ישייף. הזנו לקיר משקל ומימדים גדולים על מנת שהרובוט לא יוכל להזיז אותו.

על מנת לקבוע את הכוח בו הרובוט ישייף את הקיר, הוספנו לחוק הבקרה רכיב נוסף – וקטור כוחות בעזרתו נוכל להשפיע על איזור העבודה של הרובוט.

בשלב זה השוונו בין שני סוגי בקרת עכבה (IM ו – PB) וגם בין חוג בקרה PID פשוט.

A picture containing toy

Description automatically generated

**חוק בקרה:**

בדומה ל Peg In a Holeגם בסימולציה זו ביצענו בקרה היברידית המשלבת בין בקרת PID לבין חוק הבקרה Instantaneous Model Impedance Control , כאשר חוק הבקרה הינו PID בלבד עד רגע פגיעת החלק במשטח המשחק (אשר מזוהה באמצעות סנסור הכוח), שם חוק הבקרה מתחלף להיות Instantaneous Model Impedance Control כפי שמתואר במבוא. כך שאם המסלול ומתאר השטח ידוע במדויק ומאפשר הכנסת החלק ללא שגיאה וללא פגיעה במשטח המשחק, חוק הבקרה ישאר PID, ואילו רק במקרה של פגיעה בחלק המשטח, כתוצאה משגיאה או כתוצאה מאפיצות קרובה, חוק הבקרה יתחלף לבקרת אימפדנס המותאמת למקרים אלו.

בסימולציה זו השווינו את התוצאות אל מול שימוש בחוק בקרה PB-IC במקום ב- IM-IC.

חוק בקרה Instantaneous Model + Position Based Impedance Control:

מטריצת האימפדנס:

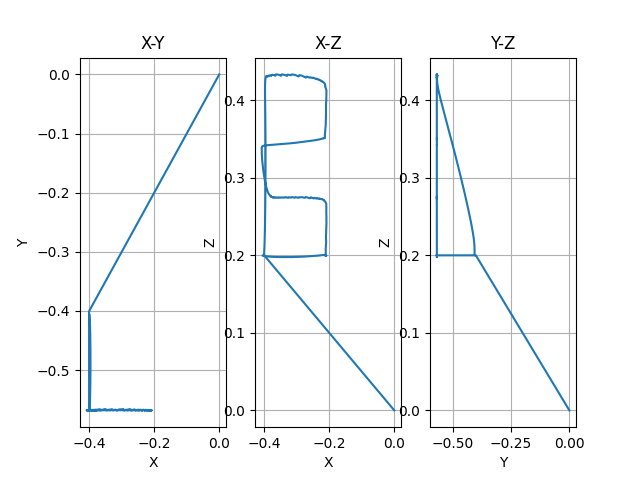
**תוצאות:**

1. בעבור התוצאות הבאות נציין כי הרובוט משייף במישור X-Z והנורמל לקיר הינו בכיוון Y.

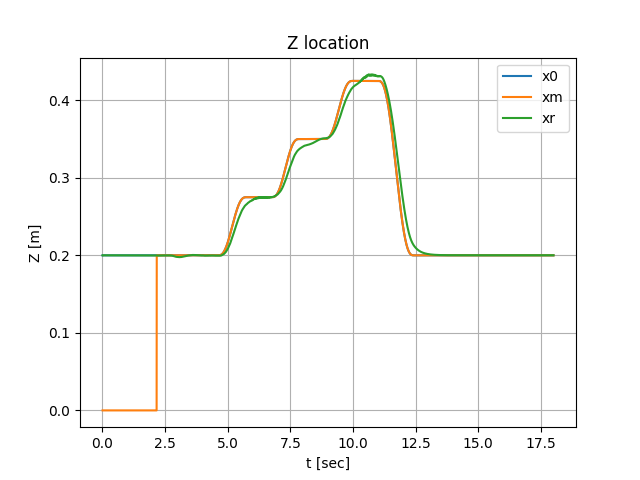
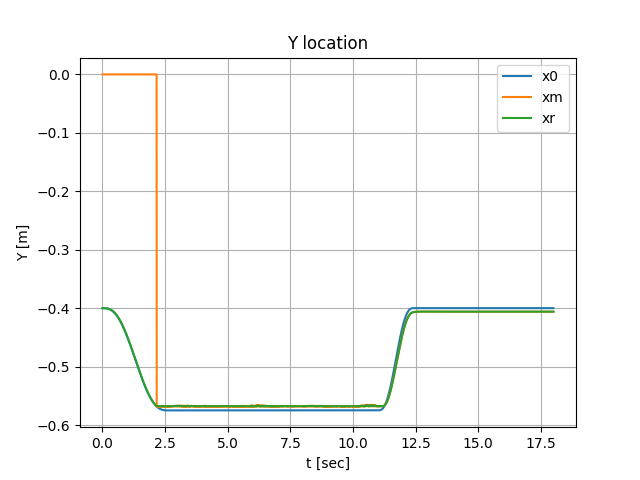
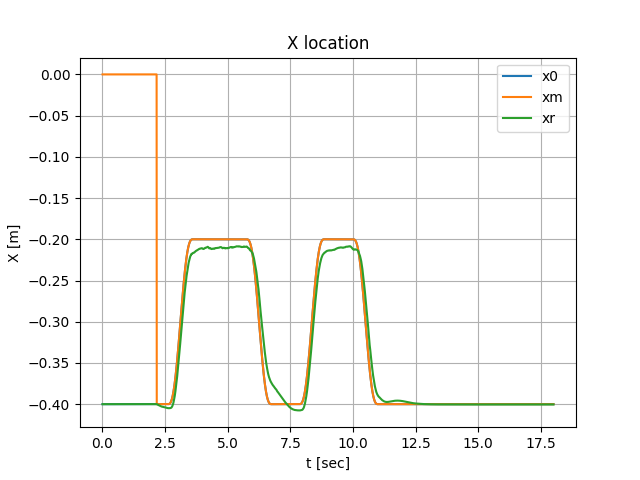
נשווה תחילה בין מסלולי יחידת הקצה של הרובוט בין שלושת חוקי הבקרה.

מודל האימפדנס IM-IC:

מסלול יחידת הקצה במרחב:



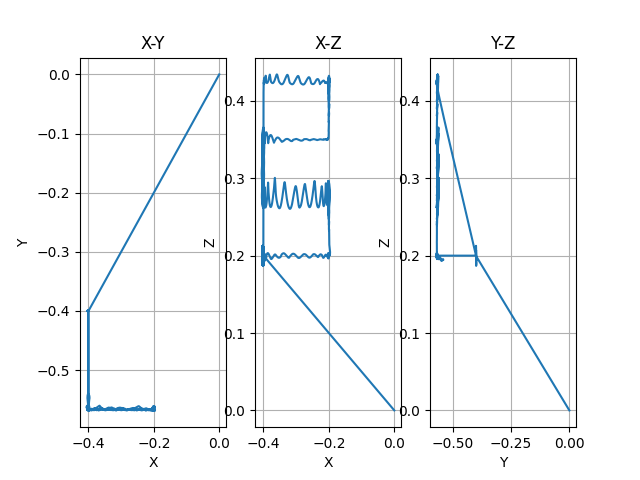
מסלול יחידת הקצה בזמן:



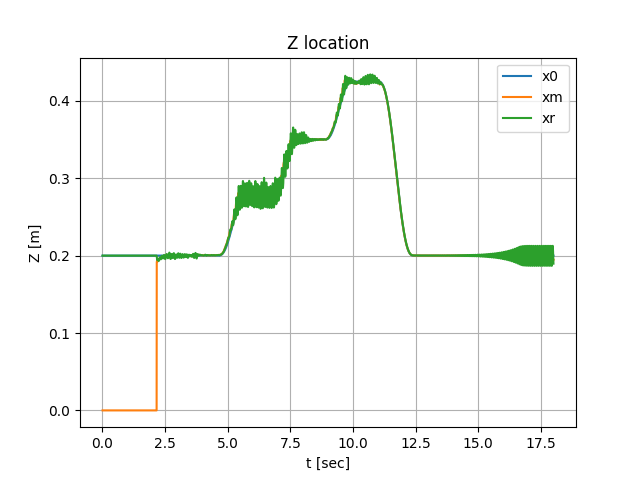
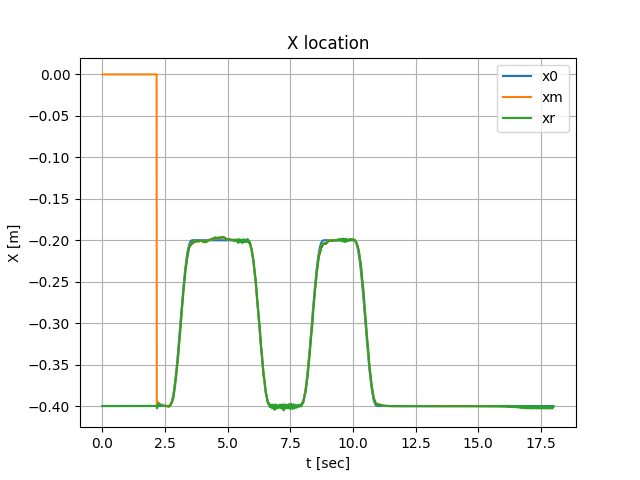
ניתן לראות כי הרובוט עוקב אחר מסלול האימפדנס עם שגיאה קלה בצירים X ו- Y, אך כמעט וללא תנודות לאורך המסלול.

מודל האימפדנס PB-IC:

מסלול יחידת הקצה במרחב:



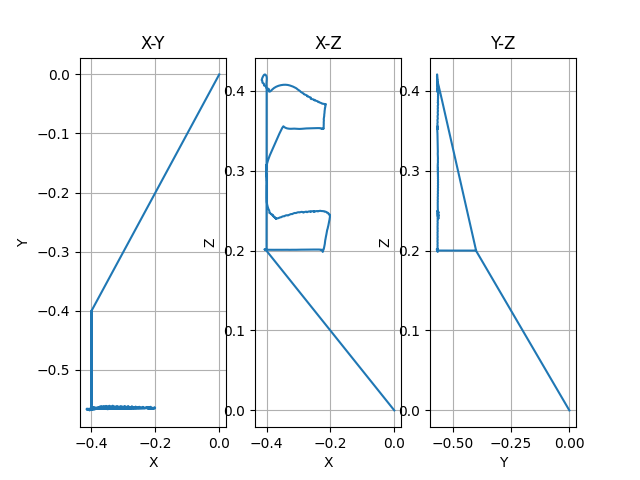
מסלול יחידת הקצה בזמן:



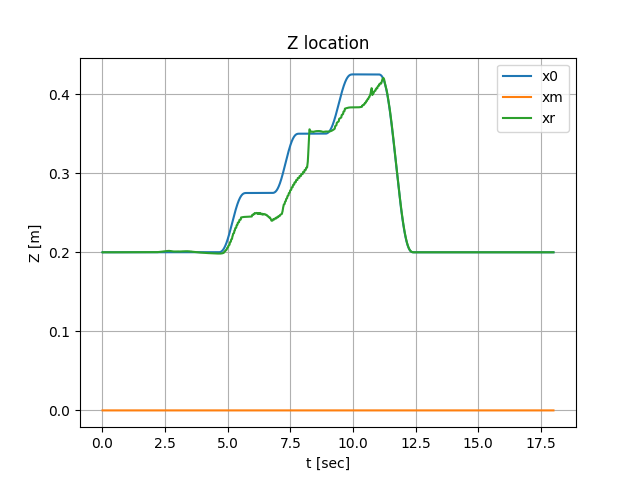
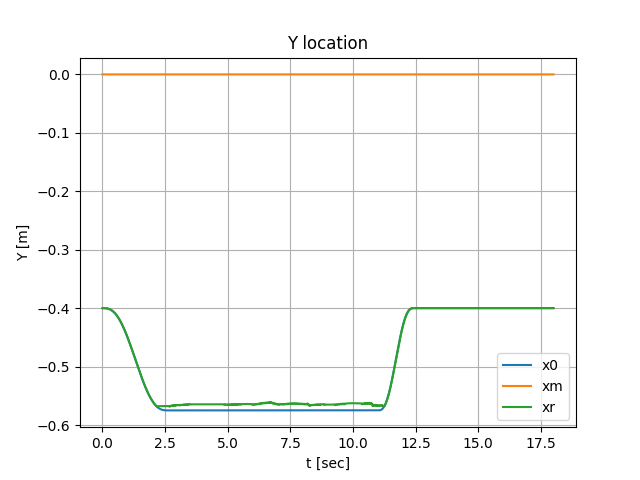
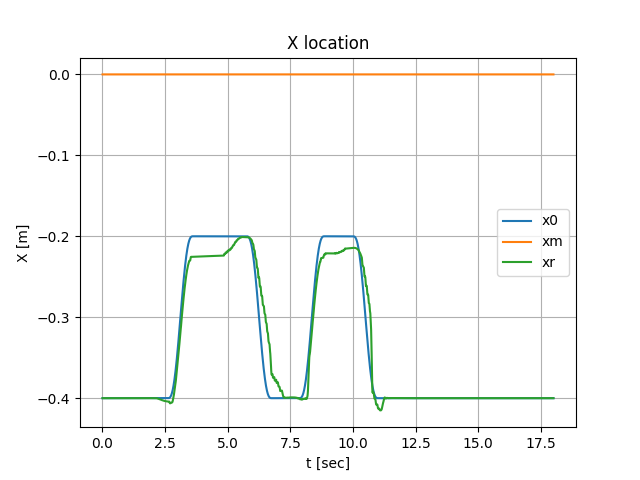
ניתן לראות מהגרפים כי שחוק בקרה זה יחידת הקצה עוקבת יותר במדויק אחר מסלול האימפדנס אך יחדת הקצה חווה רעידות רבות מאוד לכל אורך המסלול.

חוג בקרה PID:

מסלול יחידת הקצה במרחב:



מסלול יחידת הקצה בזמן:

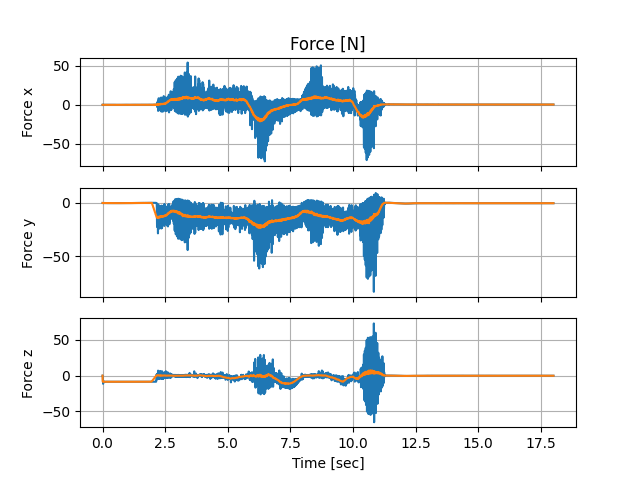
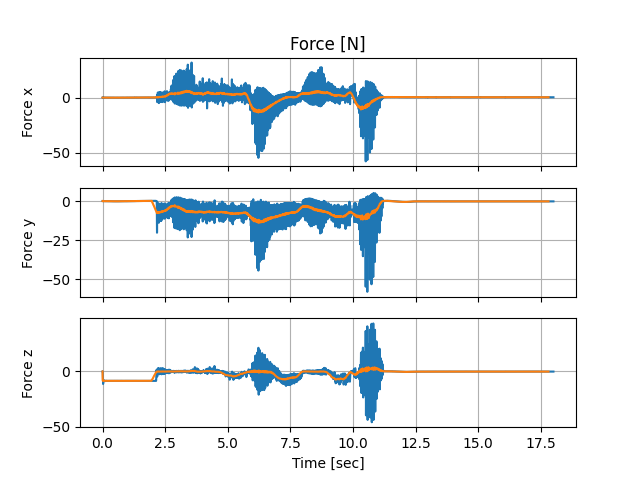


ניתן לראות כי בחוק בקרה זה הרובוט עוקב בצורה גרועה מאוד אחר המסלול.

השוואת הכוחות בין שלושת חוגי הבקרה:

מאחר ופעולות השיוף היוצרת גרירה בעלת חיכוך רב מכניסה הרבה רעש לחיישן, הוספנו לגרפים הבאים ממוצע נע של הכוחות (בכתום). חישוב הממוצע התבצע במקטעים של 100 ערכים (0.2 שניות) ומציג את ערכי הכוחות בצורה ברורה יותר ממנה אפשר להניב מסקנות.

מודל האימפדנס IM-IC:

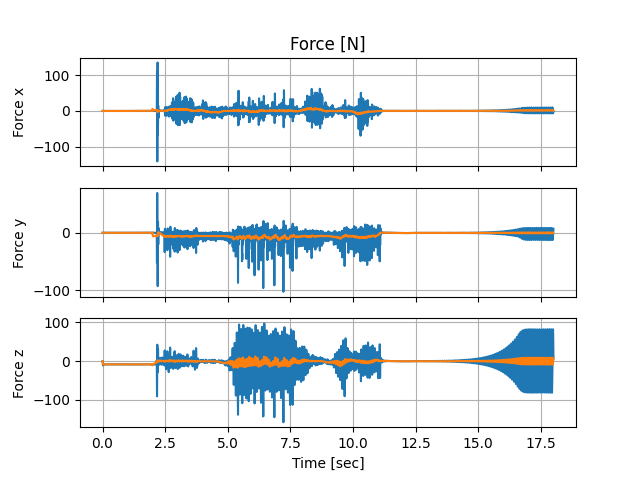


שיוף סביב 5[N]

שיוף סביב 15[N]

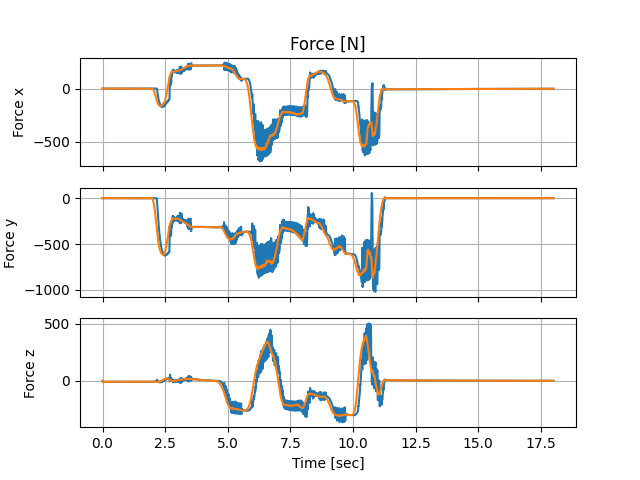
ניתן לראות כי ההבדל הניכר היחיד בין שני המצבים הוא הכוח בציר ה – Y. הגרף מתנהג בקורולציה ובאותו האופן אם כי בגדלים שונים (יש שינוי בסקאלת הגרף בציר Y).

מודל האימפדנס PB-IC:



שיוף סביב 5[N]

ניתן לראות כי הרובוט חווה כוחות רועשים הרבה יותר בהשוואה לחוק הבקרה IM-IC.

חוג בקרה PID:

ניתן לראות כי הרובוט חווה כוחות גדולים הרבה יותר בהשוואה לחוקי הבקרה בשיטות האימפדנס השונות, וכי הכוח משתנה לאורך המסלול בצורה דרסטית.

על מנת להבין את השינוי בכוח לאורך המסלול נשווה בין כוחות המינימום והמקסימום של הממוצע הנע לאורך המסלול בציר – Y :

|  |  |  |
| --- | --- | --- |
|  |  |  |
| IM 5[N] | -13.86 | 0.16 |
| IM 15[N] | -23.28 | 0.23 |
| PB 5[N] | -12.54 | 0.74 |
| PID 5[N] | -870.5 | 0.45 |

נשים לב כי ערך הממוצע בבקרת PID משתנה בצורה משמעותית יותר מאשר בחוקי האימפדנס. בנוסף, נשים לב כי ערך הממוצע דווקא בשיטת PB מתפזר בצורה פחות נמוכה משיטת הIM, למרות הרעשים הגדולים שראינו בשיטה זו. כמו כן, נשים לב כי כאשר הגדרנו שהכוח יהיה באיזור כך גם ערך המינימום השתנה בהתאם, כאשר ערך המקסימום נשאר בערך באותו התחום.

## מסקנות

ניתן לראות כי מודל האימפדנס אכן מממש את מטרתו בפרויקט זה, ובעזרתו הרובוט מסוגל להגיב לסביבה ולמתן את הכוחות אותם הוא מרגיש.

בנוסף, ניתן לראות כי למרות שהמודל PB מצליח לבצע את המשימה תוך עקיבה יותר מדויקת על המסלול ושמירה על כוחות סביב הערך שהוגדר, המסלול בשיטה זו תנודתית בצורה משמועותית וכן אינה מצליחה לרסן את רעשי הכוחות שהתפסנית חוותה.

לכן במקרים מורכבים כמו Peg In a Hole והמשייף, מומלץ להשתמש במודל ה- IM-IC.