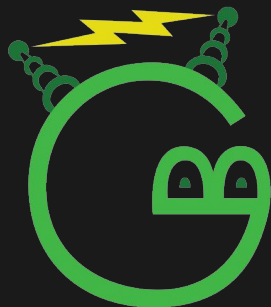


Motion Profiling

פלאג כדורי - GreenBlitz #4590





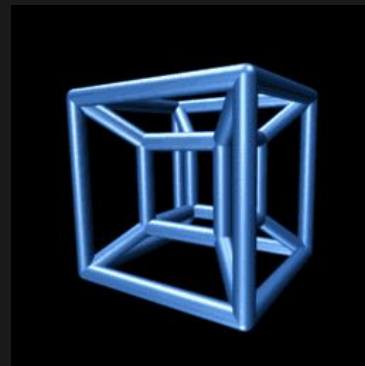
אבל רגע לפני שמתחילים...

אוטונומי?
?Motion Profiling 2D

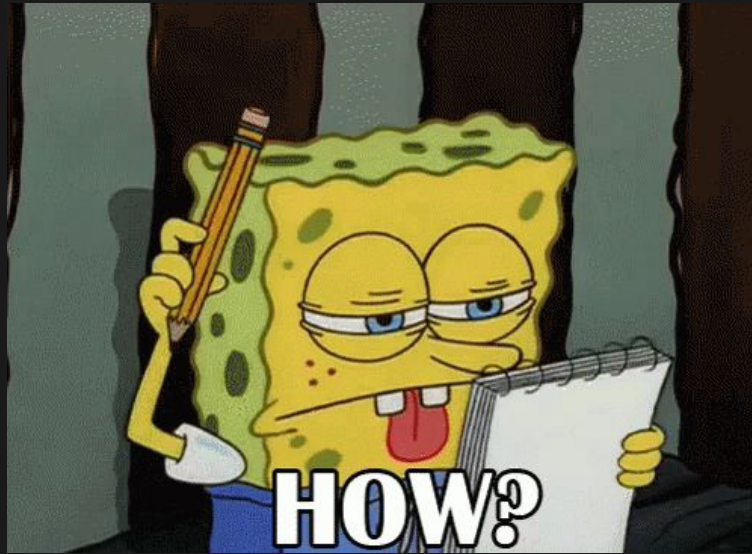
אז על מה נדבר היום?

- מה הקונספט? מה יוצא לנו מזה?
- Motion Profiling 1D
- Motion Profiling 2D

*בהרצאה היום נדבר על המימוש שלנו - GreenBlitz #4590



מה הקונספט? מה יוצא לי מזה?



- ניצור מסלול של מהירות לפי מיקום
- ממנו נחשב מסלול של מהירות לפי זמן
- נעקוב אחריו
- והנה קיבלנו אוטונומי



קצת סימונים



- v_{max} המהירות המקסימלית המותרת בקטע:

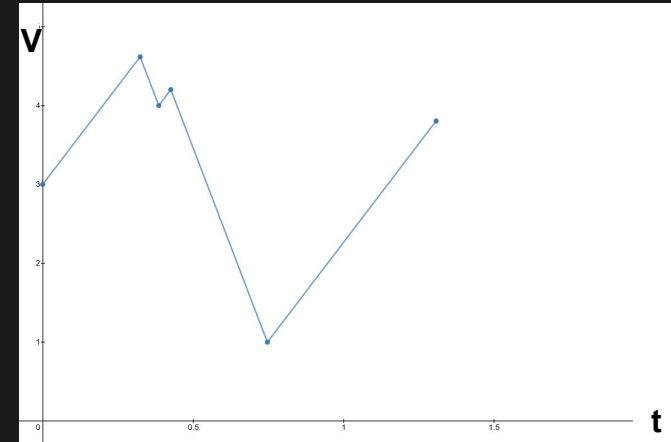
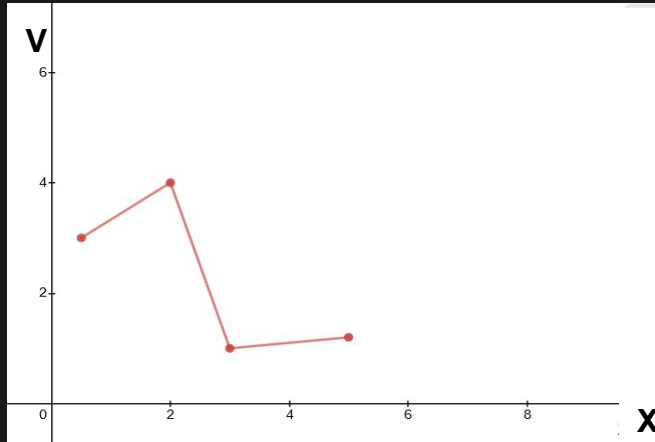
- a_{max} התאוצה המקסימלית האפשרית בקטע:

- a_{min} התאוצה המינימלית האפשרית בקטע:
זו למעשה תאוצה (תאוצה שלילית).

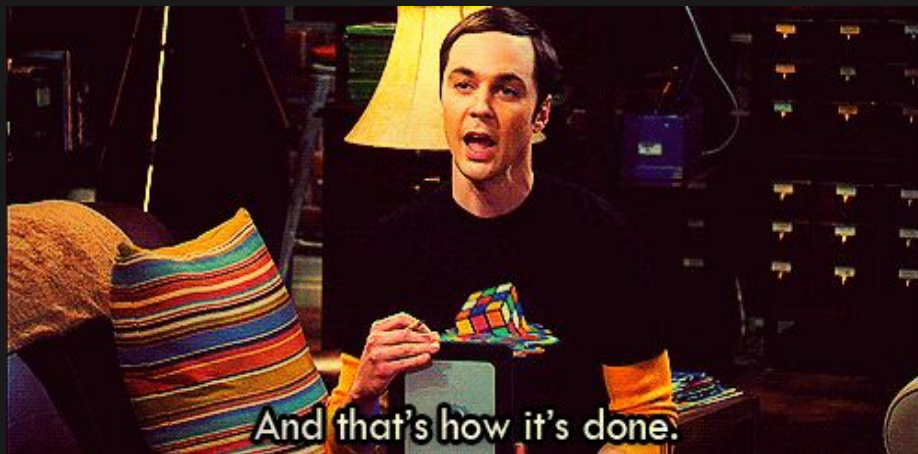
Motion Profiling 1D

המטרה הכללית:

להפוך אוסף של נקודות של (x, v) לגרף של מהירות לפי זמן
בהינתן v_{max} , a_{max} , a_{min} .



איך עושים את זה?

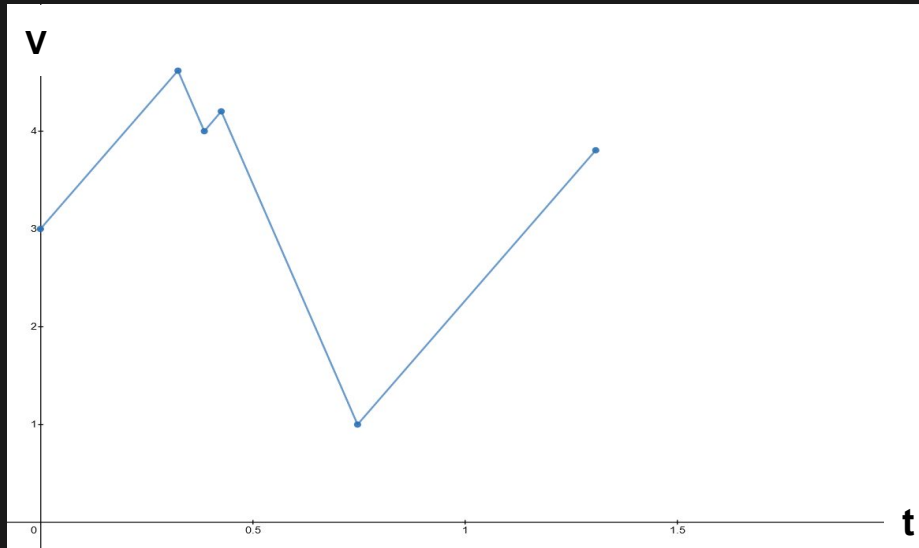


נפשט את הבעיה:

- נסתכל בכל פעם רק על שתי נקודות עוקבות מתוך האוסף
- נחשב גרף של מהירות לזמן ביניהן
- נאחד את כל הגרפים שקיבלנו
- וסיימנו (:



מה הוא גרף מהירות-זמן?



ביישום שלנו גרף מהירות-זמן מיוצג על ידי רשימה של מקטעים באורכים משתנים כאשר כל מקטע הוא בעל תאוצה קבועה.

איך מחשבים אותו?

בגישה שלנו:

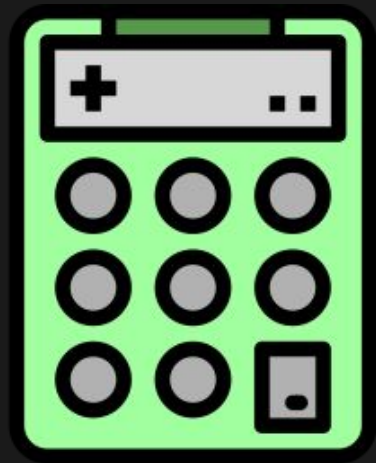
- שלב 1 (משולש מהירויות):

נחשב גרף מהירות זמן לפי מגבלות התאוצה ובהתעלמות ממגבלת המהירות.

- שלב 2 (שיטוח לטרפז):

נתאים את הגרף למגבלת המהירות אם צריך.

בשורה התחתונה - הגישה הזו תמיד תנסה להאיץ הכי מהר למהירות המקסימלית ולהאט הכי מהר.





ועכשיו לחלק הפרקטי

נסתכל על זוג נקודות עוקבות $A = (x_1, v_1); B = (x_2, v_2)$ מתוך האוסף הנתון ונמיר לגרף לפי זמן.

נרצה להאיץ כמה שנוכל ביניהן ואז להאט אם צריך כדי להגיע ל-B במהירות הנכונה.

נסמן את המרחק: $S = x_2 - x_1$

*נתייחס ל-S כחיובי כעת אך הוא יכול להיות גם שלילי.



שלב 1: משולש מהירויות

נתעלם מההגבלה של v_{max} ונבצע את הדרך הכי מהר בהתחשב במגבלות התאוצה.

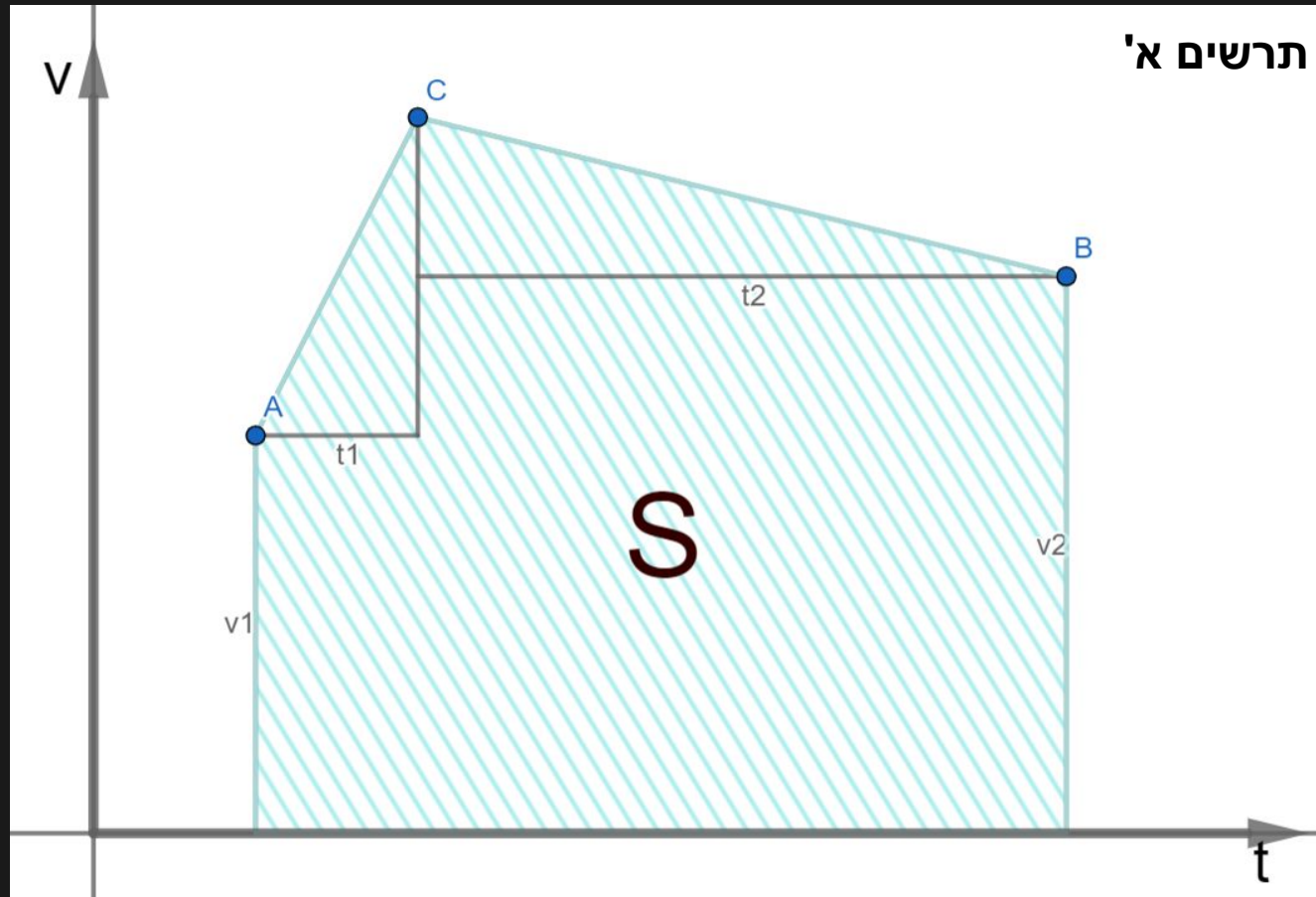
כלומר, נאיץ ב- a_{max} עד לנקודה C מסוימת ואז נאיץ ב- a_{min} .
(כלומר נאט) עד נקודה B.

נסמן:

- t_1 - כמות הזמן בה אנו צריכים להאיץ ב- a_{max} .
- t_2 - כמות הזמן בה אנו צריכים להאיץ ב- a_{min} .



תרשים א'



ומבחינה מתמטית?

כדי למצוא את t_1, t_2 נייצר שתי משוואות בשני נעלמים ונפתור.

$$v_2 = v_1 + a_1 t_1 + a_2 t_2$$

$$S = t_1 v_1 + \frac{1}{2} t_1 (a_1 t_1) + t_2 v_2 + \frac{1}{2} t_2 (-a_2 t_2)$$

המשוואה השנייה נוצרת על ידי סכימת השטחים של הצורות שמרכיבות את הטרפז.

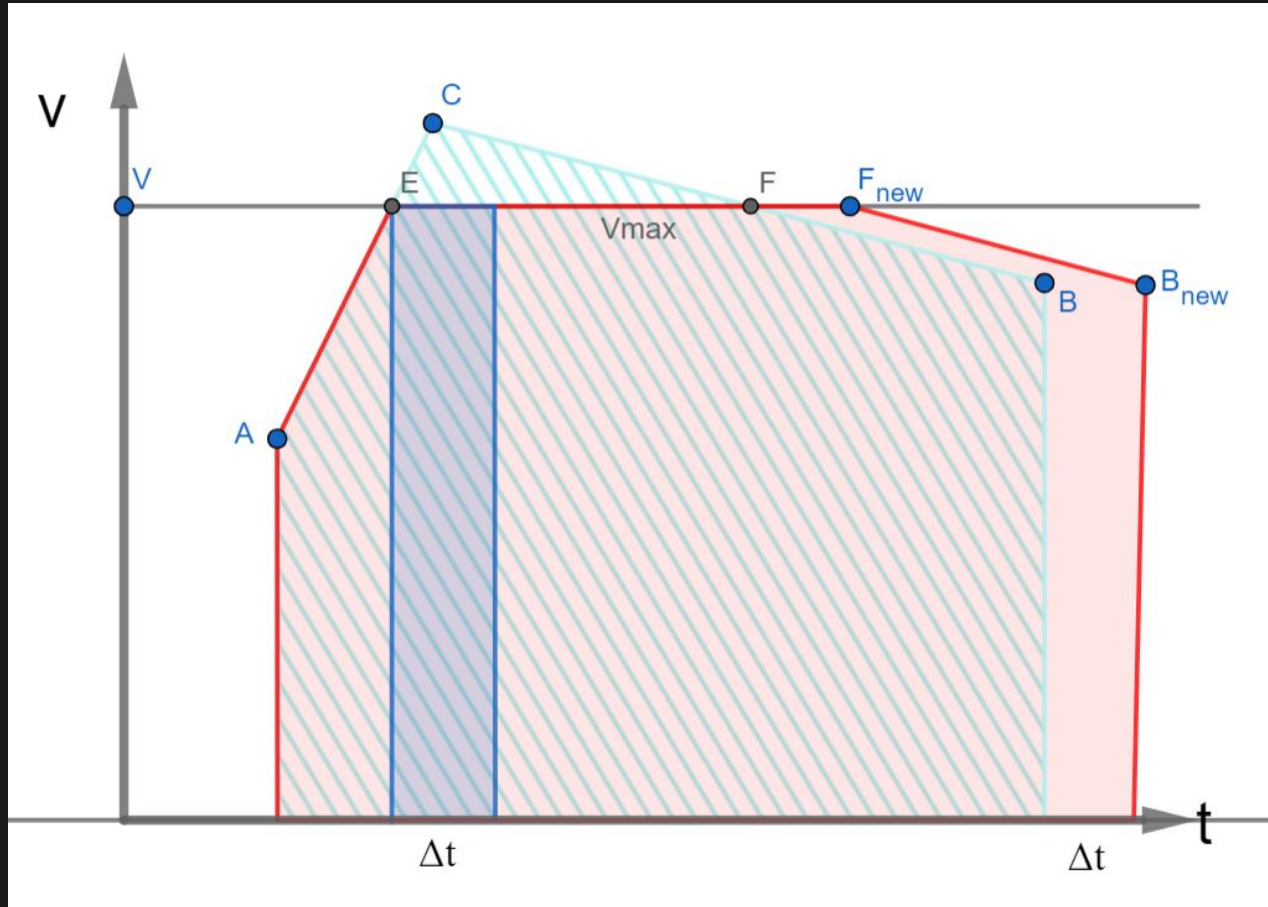




שלב 2: שיטוח לטרפז

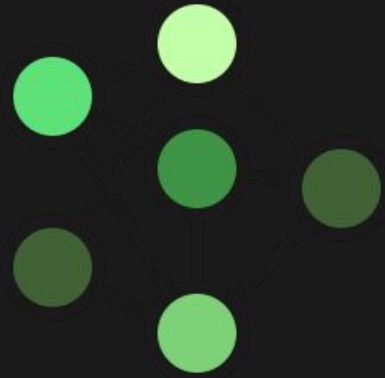
עכשיו נזכר בהגבלה של $v = v_{max}$, ונתאים את הגרף אם עברנו אותה.

למעשה נשטח את החלק העליון של הגרף שמעל $v = v_{max}$.
יש לדאוג לכך שהשטח מתחת לגרף יישאר S , זו הדרישה למרחק בין A ל-B.





שיטוח לטרפז האלגוריתם



- נחשב את נקודות החיתוך של הגרף הקיים (הירוק) עם $v = v_{max}$
 - נשטיח את הגרף
 - נחשב את S_1 - שטח המשולש שהורדנו
 - נוסיף את המלבן הכחול ששטחו S_1 ורוחבו $\Delta t = \frac{S_1}{v_{max}}$, ונזיז את הגרף ב Δt
- כך קיבלנו את הגרף החדש (האדום) שעומד במגבלת המהירות ושטחו S בדיוק כמו שרצינו!



עד כאן 1D Motion Profiling

שאלות?





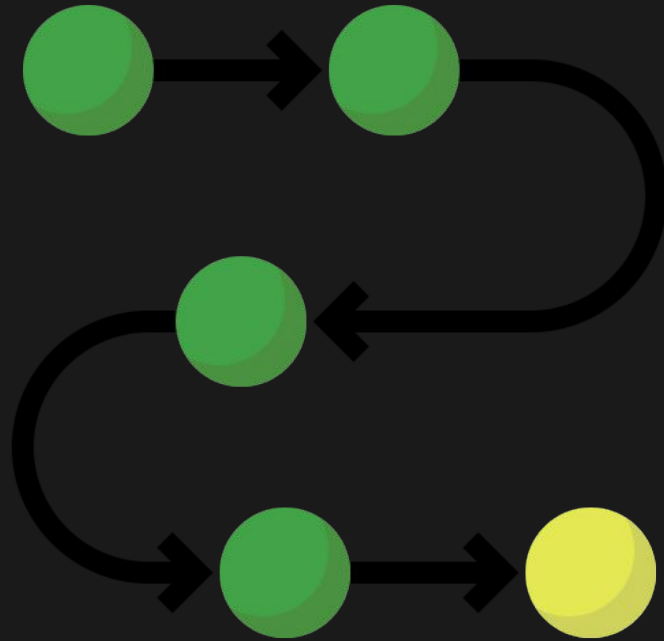
ועכשיו ל-2D Motion Profiling

אזהרה: התוכן הבא מיועד לגיל +18
למציג אין אחריות לנזק מוחי שייגרם לאחר הצפייה





Motion Profiling 2D



שלבים בתהליך:

- יצירת מסלול לאוטונומי.
- פישוט הבעיה ל-Motion Profiling 1D.
- חישוב גרפי מהירות-זמן.
- מעקב בזמן אמת אחרי הגרפים האלו.



מהירות לינארית (V)

קצב שינוי המיקום של הרובוט.
כלומר, מה המרחק שעובר הרובוט בזמן מסוים.

מהירות זוויתית (W)

קצב השינוי של זווית הרובוט.
כלומר, כמה משתנה זווית תנועת הרובוט בזמן מסוים.

תאוצה לינארית

קצב השינוי של המהירות הליניארית

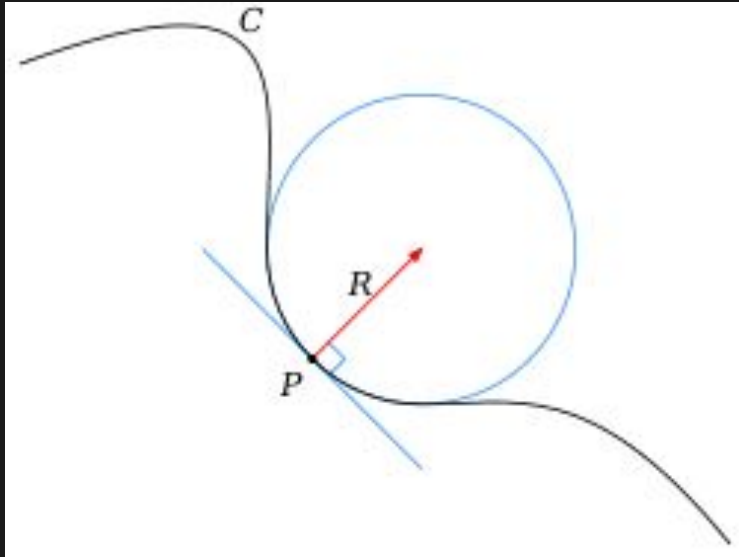
תאוצה זוויתית

קצב השינוי של המהירות הזוויתית

קצת הגדרות



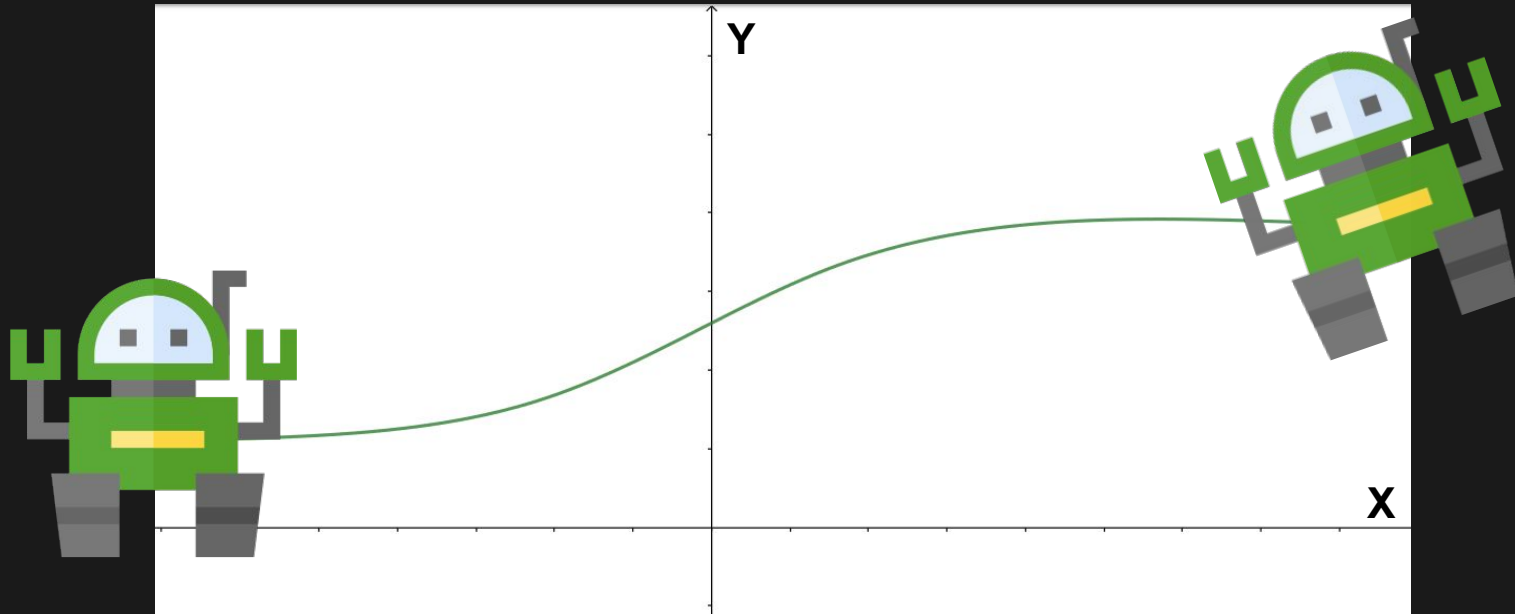
הגדרה - עקמומיות



- עקמומיות המסומנת באות K , מתארת כמה עקום המסלול, כמה רחוק הוא ממצב ישר.
- עקמומיות מוגדרת ($K = \frac{1}{R}$) כאשר R הוא הרדיוס של המעגל שעליו נוסע הרובוט בנקודה:
- לא נוכיח אבל מתקיים $K = \frac{W}{V}$.



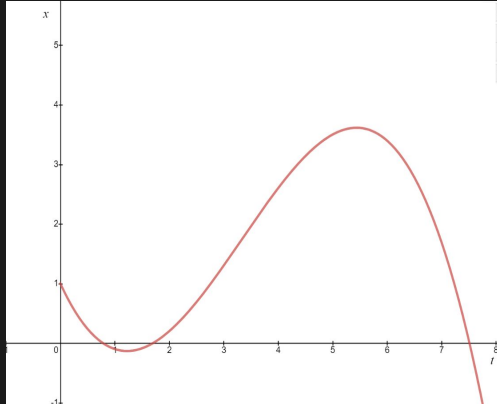
מה הוא מסלול לאוטונומי



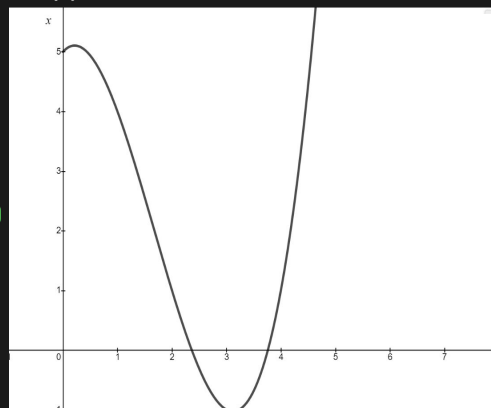
איך מסלול כזה מיוצג

נפשט את המסלול בכך שנפרק אותו לשני גרפים לפי זמן:

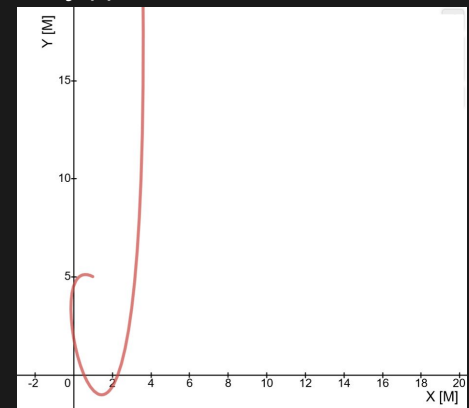
$y(t)$



$x(t)$



$x, y(t)$



איך מסלול כזה מיוצג

המשך הפשטה:

לשם הנוחות נגדיר את שני הגרפים מהשקופית הקודמת להיות פולינומים מדרגה שלישית מהצורה $ax^3 + bx^2 + cx + d$.

כלומר כל מסלול מוגדר על ידי 8 פרמטרים. (4 ל-x ו-4 ל-y). כדי ליצור מסלול נצטרך למצוא את הקבועים שמתאימים לתנאי ההתחלה והסיום שנתונים.



איך מסלול כזה מיוצג

בעיה במושגים שלנו:

אי אפשר ליצור גרפים לפי הזמן כי בתחילת התהליך אנחנו לא יודעים את הזמן של כל נקודה.

פתרון קונספטואלי:

- נחשוב על משתנה דמוי זמן t שמייצג את הסדר הכרונולוגי.
- נגדיר את ה- x וה- y להיות פונקציות לפי t במקום לפי הזמן.

כך נקבל מסלול לפי t שנוכל לעבוד איתו.



יצירת מסלול - Spline Interpolation

נתונות לנו שתי נקודות $A = (x_A, y_A, \alpha)$; $B = (x_B, y_B, \beta)$ בעלות מיקום וזווית של הרובוט בהן.
נרצה ליצור מסלול בין A ל-B.
כלומר, למצוא את שמונת הפרמטרים שמגדירים את המסלול.

כדי לעשות זאת, ניצור משוואות שיעזרו לנו לחשב את הקבועים.
כלומר, ניצור מערכת משוואות מהנתונים הקיימים ונפתור אותה.





יצירת מסלול - Spline Interpolation



משוואות לדוגמא שניתן לייצר הן:

$$x(0) = x_A \quad \bullet$$

$$y(0) = y_A \quad \bullet$$

$$x(1) = x_B \quad \bullet$$

$$y(1) = y_B \quad \bullet$$

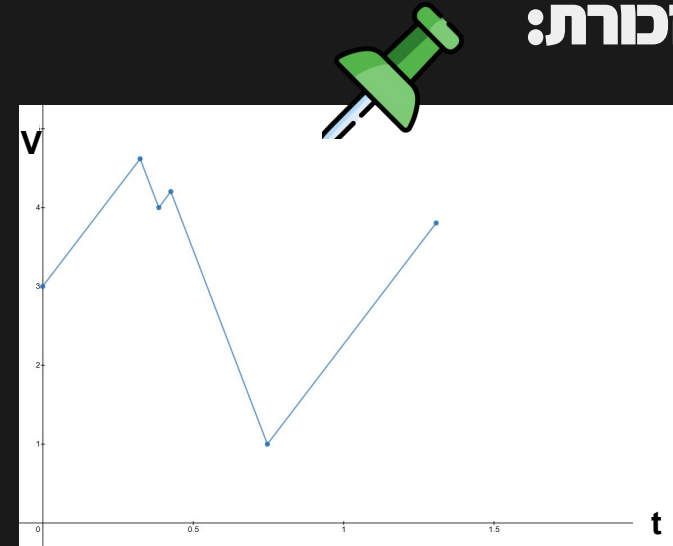
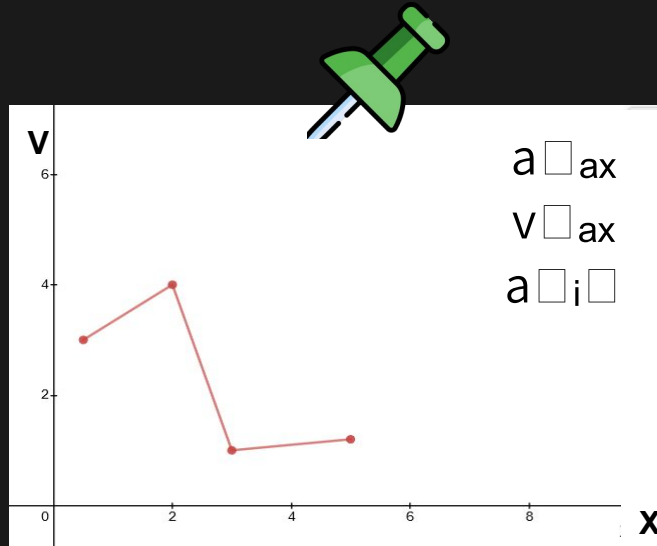
$$\frac{y'(0)}{x'(0)} = \tan \alpha \quad \bullet$$

$$\frac{y'(1)}{x'(1)} = \tan \beta \quad \bullet$$

*קיימת גרסה יותר מסובכת ומדויקת של התהליך שמשתמשת בפולינומים ממעלה 5

פישוט הבעיה ל-1D Motion Profiling

תזכורת:



- מה נקבל?
- מה עוזר לנו מהירות לינארית בכל זמן?



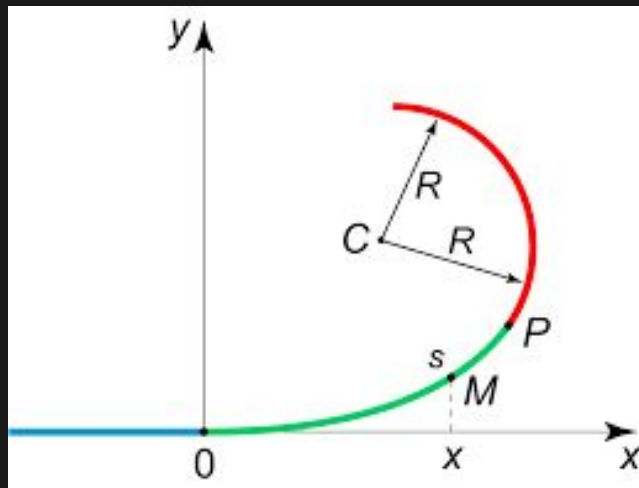
פישוט הבעיה ל-1D Motion Profiling

תהליך:

- חלוקת המסלול למקטעים.
- חישובים של v_{\max} , a_{\max} , a_{\min} לינארים בכל מקטע.
- יצירת גרף מהירות לינארית לפי מיקום.
- ביצוע התהליך של 1D Motion Profiling על המהירות הלינארית.

חלוקת המסלול למקטעים

נחלק את המסלול למקטעים קטנים יחסית.
נתייחס לכל מקטע כקשת של מעגל.



לקשת של מעגל יש עקמומיות קבועה
כי יש לה רדיוס קבוע ($K = \frac{1}{R}$).
לכן לכל מקטע יש עקמומיות קבועה K .

נחשב את עקמומיות זו בכל מקטע.



חישוב a_{min} , a_{max} , v_{max} בכל מקטע

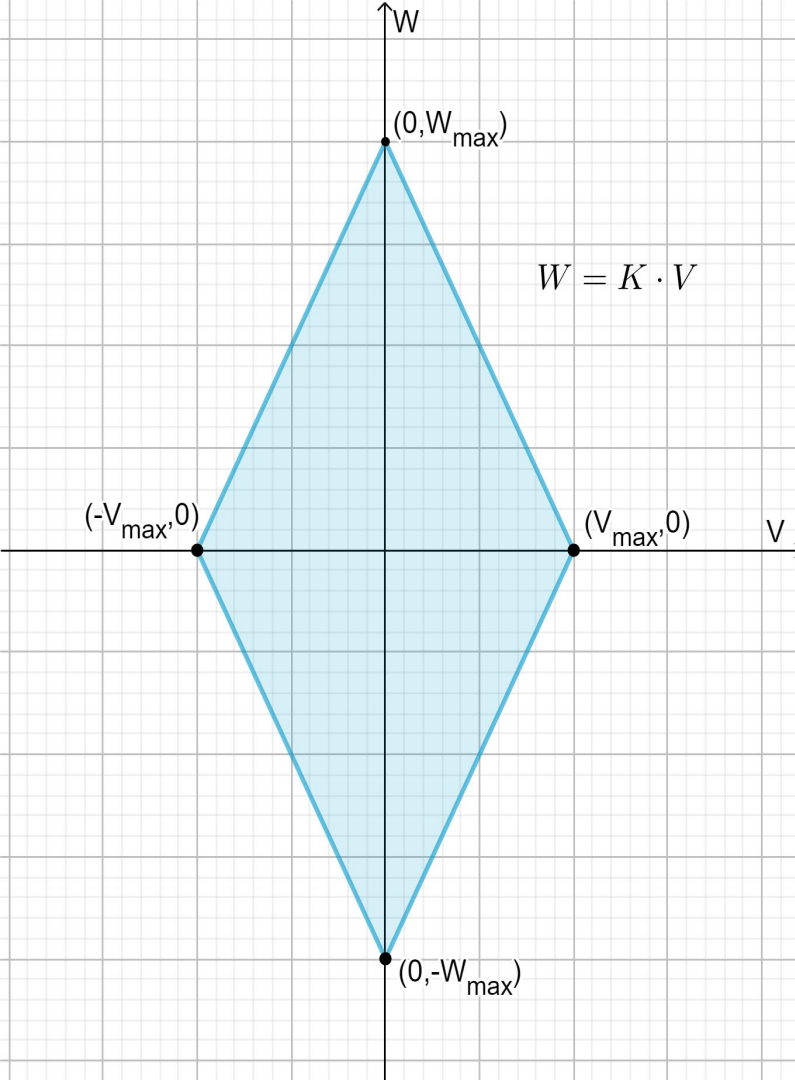
- נסתכל על מקטע מסוים בעל עקמומיות K .
- נחשב את המהירות הלינארית המקסימלית האפשרית במקטע.

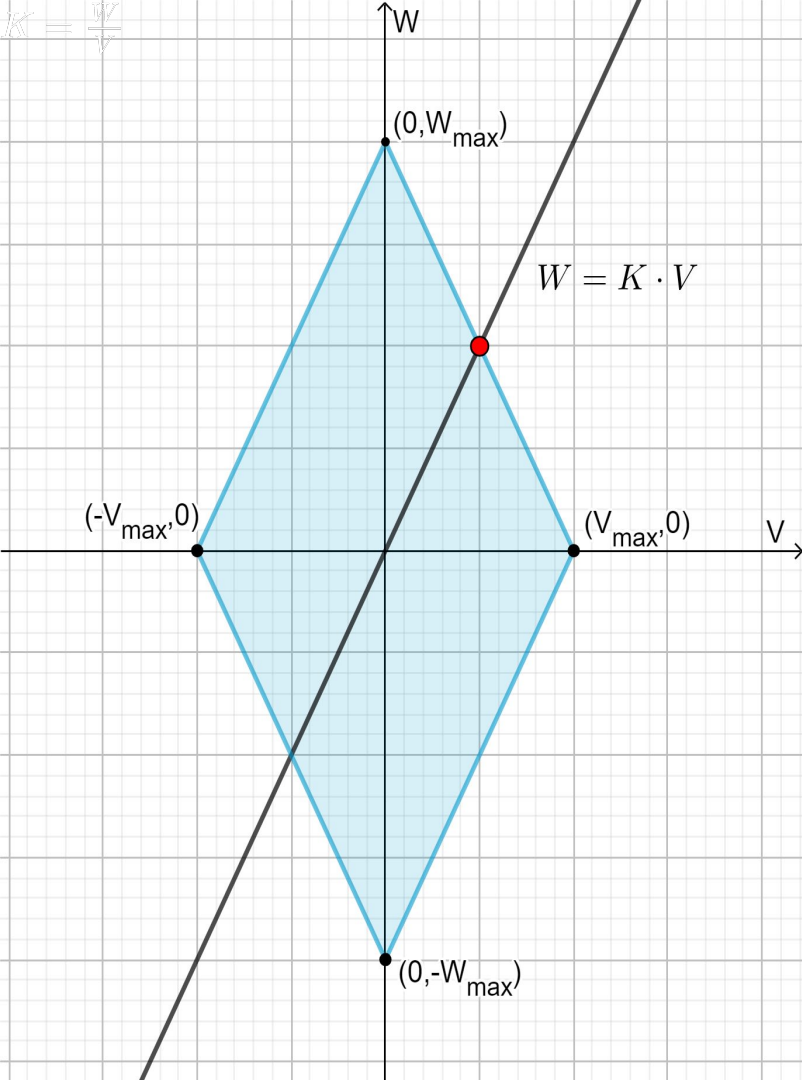
נראה במצגת רק את החישוב של v_{max} .
החישוב של a_{max} ו- a_{min} מתבצע באופן דומה.



שיטת המעוין

המעוין הצבוע מייצג את כל הקומבינציות האפשריות של מהירות לינארית וזוויתית ברובוט.





שיטת המעוין

נסתכל כעת על מקטע בעל עקמומיות K .
נרצה לדעת מה המהירות הלינארית המקסימלית
 v_{\max} שנוכל לקבל.

ראינו כי $K = \frac{W}{V}$ ולכן מתקיים $W = K \cdot V$
זה הישר המשורטט שמייצג את הקומבינציות
האפשריות של המהירויות במקטע.

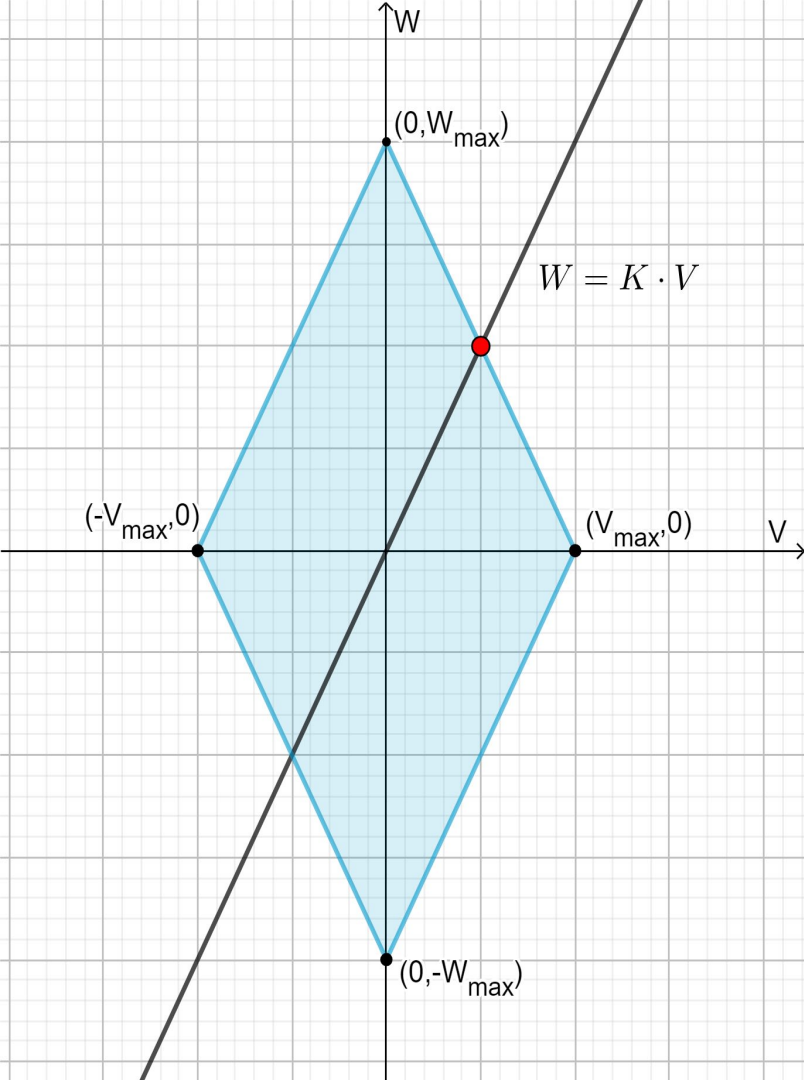


שיטת המעוין

- נחשב את שיעורי הנקודה האדומה.
- שיעור ה-x של הנקודה הזו הוא המהירות הלינארית המקסימלית בקטע.

שיטת המעוין עבור תאוצות תניב את:

- a_{\max}
- a_{\min}



יצירת גרף מהירות לינארית לפי מיקום - מה הכוונה?



ניצור גרף המתאר את המהירות הלינארית של הרובוט
לפי הדרך על המסלול שהרובוט עבר.



יצירת גרף מהירות לינארית לפי מיקום

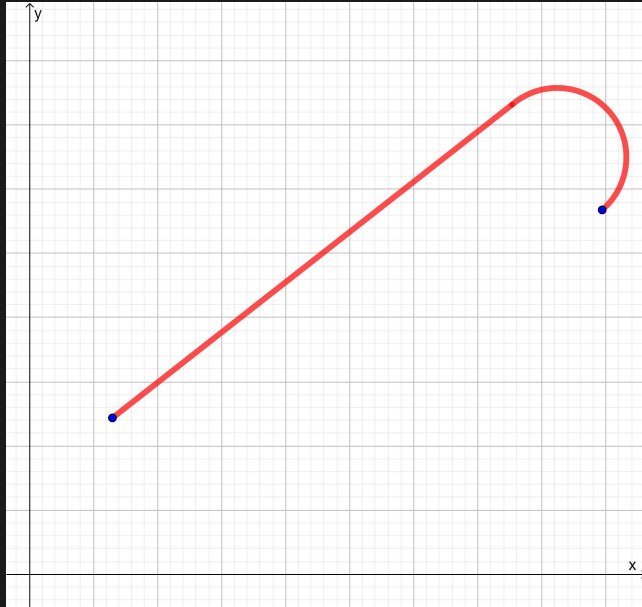
מה צריך לחשב עבור כל מקטע:

- אורך המקטע על המסלול*.
- המהירות הלינארית שנרצה בתחילת המקטע ובסופו.

זו עבודה קשה.

*נזכור שהמקטע הוא בקירוב קשת לכן נחשב את אורך המקטע כפי שמחשבים אורך של קשת.

חישוב המהירות הליניארית הרצויה בכל מקטע



באופן כללי נרצה שהמהירות הזו תהיה
הכי קרובה שאפשר למהירות המקסימלית
אבל חובה להיזהר שלא ניצור מצב לא אפשרי.

לכן תמיד צריך לבדוק שהרובוט יכול
להאיץ ולהאט בזמן.



הפעלת Motion Profiling 1D

נפעיל את תהליך Motion Profiling 1D על תוצאות החישובים האחרונים ונקבל גרף של מהירות לינארית לפי זמן.



הערה: נשים לב שערכי הקבועים v_{ax} , a_{ax} , a_{i} משתנים ממקטע למקטע ולכן נצטרך להפעיל 1D על כל מקטע בנפרד ולחבר את הגרפים שיצאו לנו אחד אחרי השני.

חישוב גרפי מהירות-זמן

יש לנו גרף של מהירות לינארית לפי זמן ונרצה לקבל גם גרף של מהירות זוויתית לפי זמן.

$$W = K \cdot V$$

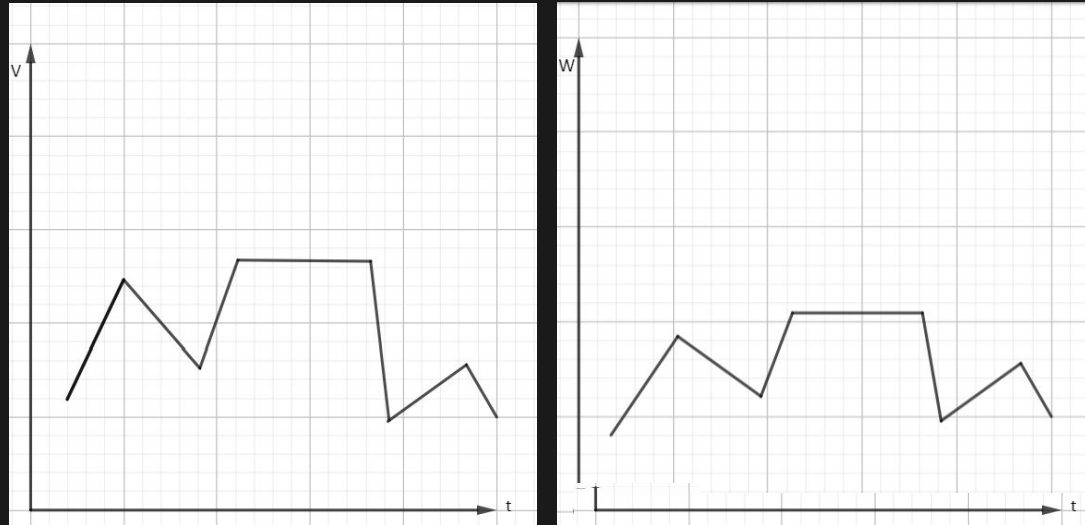
נזכור שבכל מקטע בעל עקמומיות K מתקיים

לכן, בכל מקטע נכפיל פי K את המהירות הלינארית לקבלת מהירות זוויתית באותו מקטע, וניצור גרף של המהירות הזוויתית לפי זמן.



חישוב גרפי מהירות-זמן

כעת מה שיש לנו הוא:



לשני הגרפים האלו ביחד נקרא פרופיל.

מעקב אחרי הפרופיל

בזמן אמת נעקוב אחרי הגרפים של
המהירות הלינארית והמהירות הזוויתית.
כיצד?

בכל נקודת זמן נחשב, בעזרת המהירות
הזוויתית והלינארית, את המהירות והתאוצה
שאמורות להיות לכל גלגל.

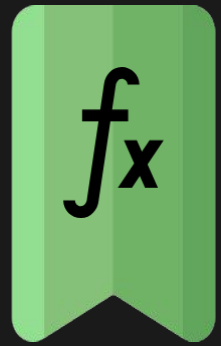


אפשרי. תבדקו בבית :)
רמז: קשור למרחק בין הגלגלים





נוסחת המנוע מהי?



נוסחת המנוע -
$$P = K_v \cdot v + K_a \cdot a$$

P - הכוח שאנחנו צריכים לתת למנוע (בין 1- לבין 1).

K_v - מקדם מהירות הרובוט*.

v - מהירות הרובוט.

K_a - מקדם תאוצת הרובוט*.

a - תאוצת הרובוט.

*קבועים שמחשבים בעזרת מדידות על הרובוט.



נשתמש בנוסחת המנוע

$$P = K_v \cdot v + K_a \cdot a : \text{נוסחת המנוע:}$$

בכל נקודה בזמן נציב את התאוצה והמהירות הרצויות
ונקבל את הכוח שצריך לתת לכל מנוע!

וסיימנו עם התיאוריה!
באופן פרקטי, צריך להוסיף PID Controller שמוודא שהרובוט
אכן עוקב אחרי גרף המהירות כמו שצריך.



כאן נגמר סיפורו של פרופיל...

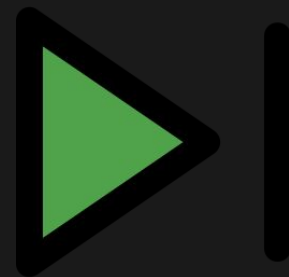
שאלות?





אופציות להתפתחות

- Live Motion 2D Profiling
- שילוב עם Adaptive Point Pursuit Controller
- גישות אחרות ל-1D Motion Profiling



תודה על ההקשבה

ומקווה שלא נשרף לכם המוח...





מקורות בשבילכם

המצגת הזו:

<https://docs.google.com/presentation/d/1VRsLZbWEF0wr2cETQHo3uqB9VYYF1XRYmkYRPqUqkSE/edit?usp=sharing>

הקוד שלנו ב-GitHub:

<https://github.com/GreenBlitz/MotionControl/tree/master/motion/src/main/java/org/greenblitz/motion/profiling>

פרטי קשר:

פלג כדורי 058-7994455, מוזמנים ליצור קשר על כל שאלה!



פיתוח:

- אלכסיי שפובלוב, בוגר #4590
- אור קרל, בוגר #4590
- אודי מטלון, בוגר #4590
- פלג כדורי, ראש צוות תוכנה #4590

תוכן:

- פלג כדורי, ראש צוות תוכנה #4590
- רואי נבו, ראש צוות תוכנה #4590

עריכה:

- אור שמר חיים, ראשת צוות בנייה #4590
- אוראל אברהם, חבר צוות תוכנה #4590
- איתי גיל, אחראי אסטרטגיה #4590
- ניצן ברסלר, ראש צוות מערכות #4590
- איתמר אוריין, ראש צוות פרסים ויח"צ #4590

קרדיטים

