

Black

Red

White

Yellow

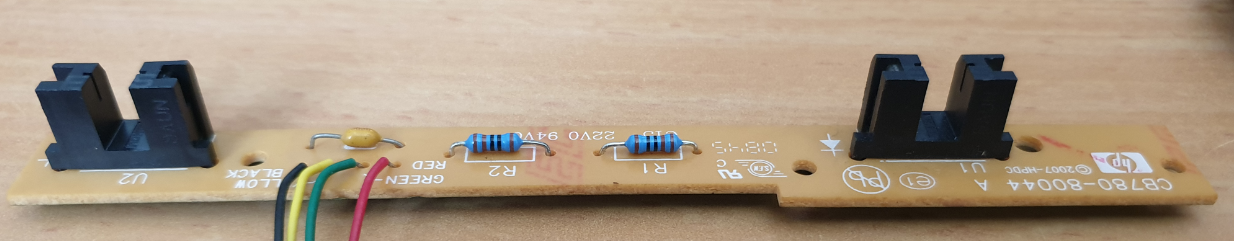
Orange

Green

חיבורים

1. M1, M2 – אספקת המתח למנוע
2. VCC, GND Holzer – אספקת מתח ל- Encoder (עובד ב- 3.3V וגם ב- 5V. מתח היציאה בהתאם. אני השתמשתי ב- 5V). Holzer מצא את Effect Hall לפני Hall
3. A, B שני גלאים ב 90 מעלות כדי לדעת את כיוון הסיבוב (בודקים אם A מקדים את B או הפוך)

ה- U-Sensor שימש למדידת סיבוב מלא של הגלגל כדי למצוא את מספר הפולסים שיש לגלאים A או B בסיבוב שלם



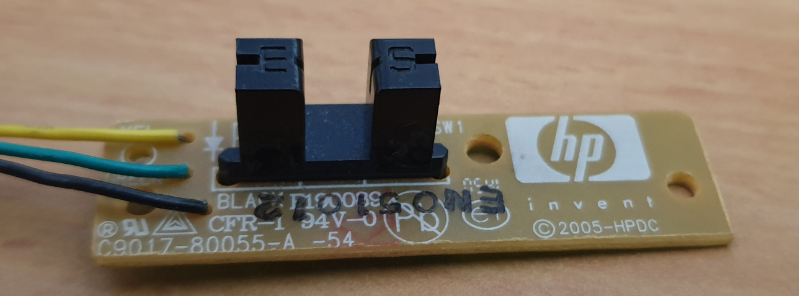
אדום – 5V

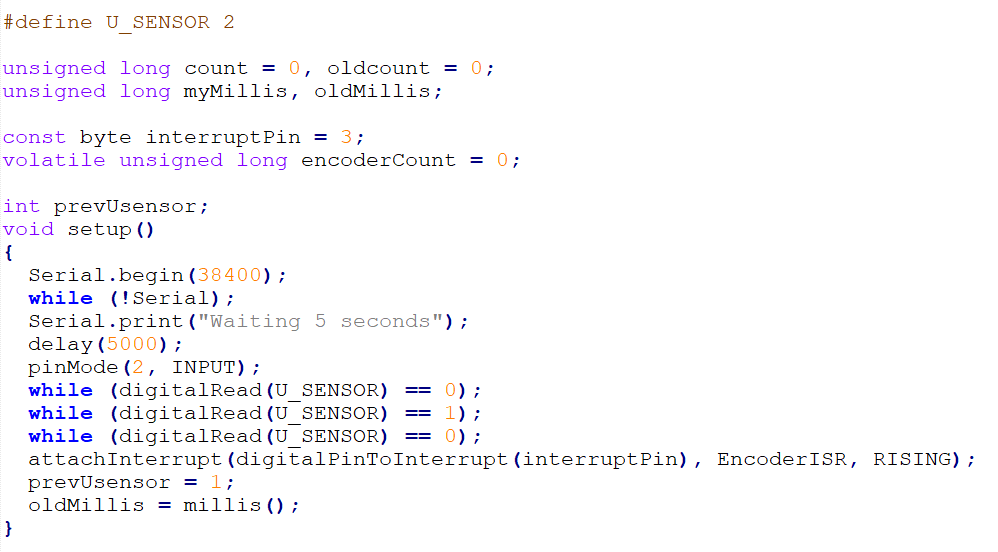
שחור – GND

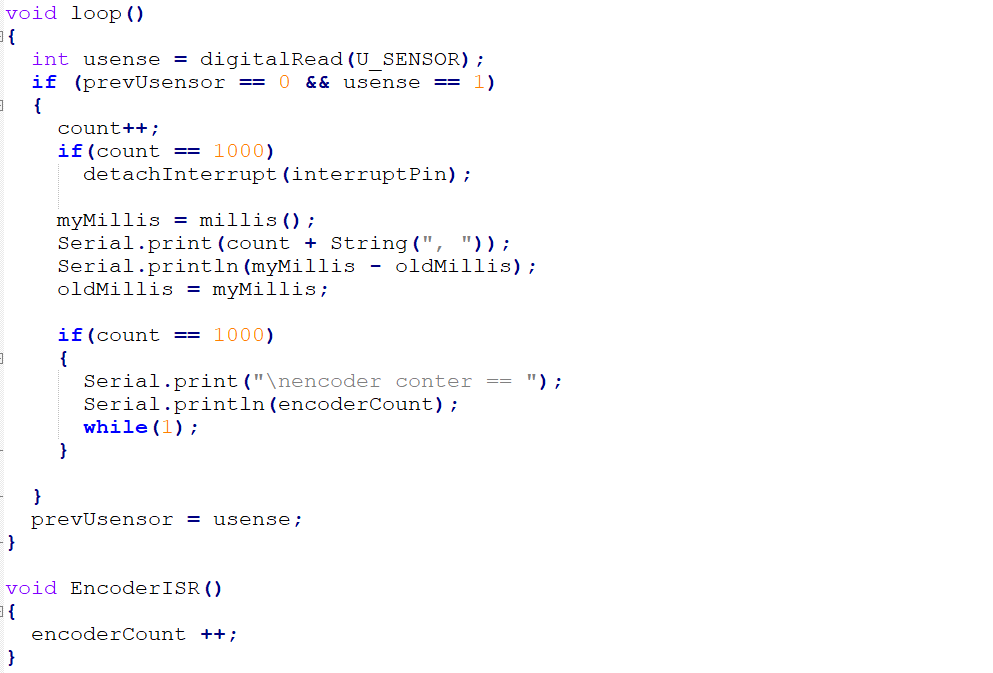
ה- LED המשדר מחובר טורית למתח דרך נגד ה- 110 אוהם (אחד לכל LED ), הקולקטור של הטרנזוסטור הפוטואלקטרי מתחבר לחוט הירוק (והשני לצהוב) והאמיטר מחובר לאדמה.

יש לחבר את החוט הירוק (והצהוב) ל VCC דרך נגד של 5K לפחות (300 אוהם לא מכניס אותו לרוויה)

במודול הבא הרעיון דומה אבל יש לספק את כל הנגדים בהתאם למודול הקודם



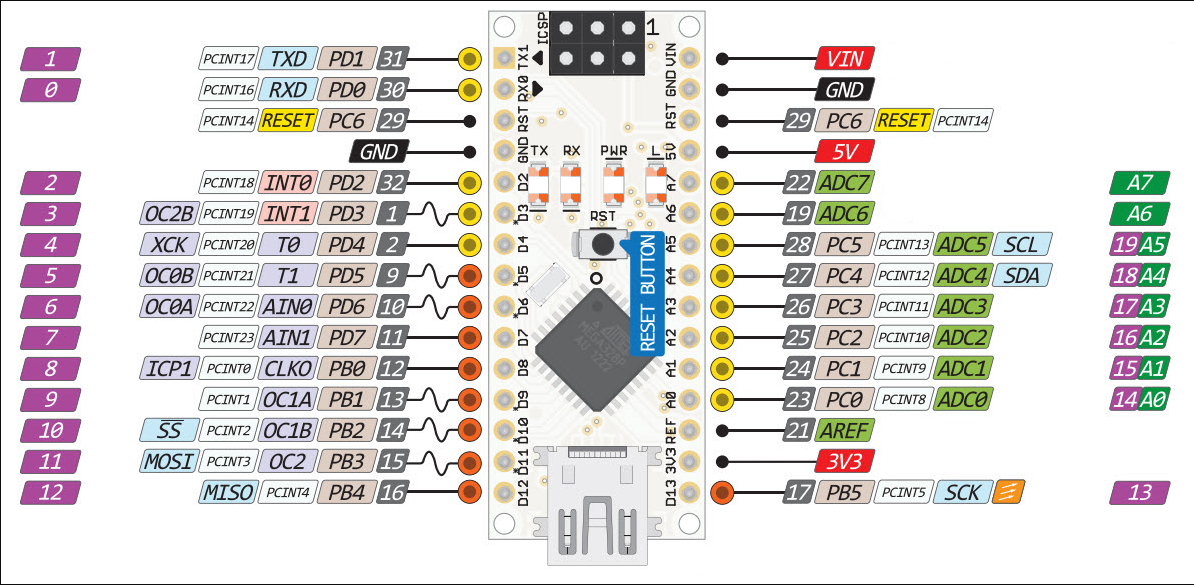




**אספקת מתח ל- UNO**

<https://www.technobyte.org/2016/07/power-up-the-arduino-uno/>

Nano pinout



**המהירות הקווית יחסית למספר המניות בשנייה של המקודד**

R – רדיוס הגלגל ב- ס"מ

ω – המהירות הזוויתית של הגלגל ב- רדיאנים לשנייה

V – מהירות הקווית ב- ס"מ לשנייה

E – מספר המניות של האינקודר לסיבוב שלם

הקשר בין מספר מניות n לבין זווית הסיבוב של הגלגל α

µ - מספר מניות לשנייה של האינקודר

נמצא את המהירות הקווית יחסית למספר המניות לשנייה

ראשית נמצא את הקשר בין µ ל- ω

וכעת נמצא את הקשר בין המהירות הקווית ל- µ

המהירות המקסימלית של המנוע היא 150rpm שהם

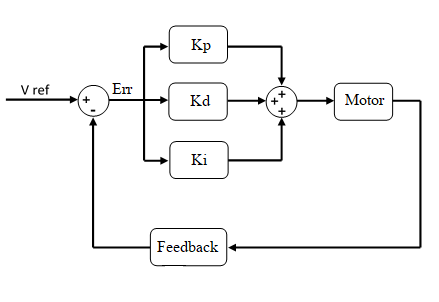
נחשב את מספר המניות לשנייה יחסית למהירות הקווית

ברובוט שלי

R = 6.45 / 2 = 3.225 cm

E = 824.424

נחשב את עבור מהירויות של 1 , 3, 10, 15 ונקבל בהתאמה

נבדוק ונחשב את המהירות בכל 50ms, נמצע שתי קריאות אחרונות כלומר, נחשב מהירות ממוצעת של חלון של 100ms שבכל פעם נע 50ms קדימה

**גישה חדשה ( ב- 03-BalancingTest )**

בגישה הקודמת ה- PID שלט על מהירות הגלגלים (כל גלגל בנפרד) בטור לזה בנינו PID נוסף ואיזון בעזרת ה- 6050. על ידי כך שהסענו את הגלגלים במהירות מתאימה לכיוון הנפילה וכך ייצבנו את המערכת.

הבעיה – המערכת קיבלה מהירות קבועה כל שהיא בזמן הייצוב ושמרה עליה, המהירות יכולה להתגבר או להתבטל בהתאם לתיקון הנוסף, אם הרובוט לא מאוזן מתחילה נסיעה מטורפת לכיוון מסוים (תמיד לאותו הכיוון) עד שאין ביכולת המנועים לאזן (מעל מהירות מסוימת לא ניתן להאיץ במגמת התנועה).

הפתרון – נבנה מערכת שתפקידה לאזן את הרובוט ובטור לה נבנה מערכת נוספת שתפקידה לשמור על מהירות קבועה של הגלגלים על ידי תיקון הטיה עדין של הרובוט לכיוון המתאים וכך נמנע מהתפתחות תנועה לא מכוונת.

מאחר וניתן למדוד מרחקי תנועה אפשר אפילו לתקן את המיקום של הרובוט שלא יתחיל להתהלך באופן לא רצוי.

**יש לחשוב כיצד לשמור על Yaw קבוע**

בקרת המהירות תתבצע בשתי שלבים

1. אם המהירות 0 נקרא לייחוס את האינקודרים ואם יתרחק משם נחזירו על ידי הטיה , בשלב הבא יש לשמור גם על yaw קבוע
2. עם המהירות גבוה מ- 0 נמדוד את מהירות הנסיעה ונבקר אותה בעזרת הטיות

**רעיון לאיזון עצמי**

בהדלקה, אם הרובוט לא מאוזן תתחיל תנועה לכיוון מסוים. מערכת תיקון אוטומטי תחזיר אותו למקום ותטה את הייחוס לכיוון השני וכך בחיפוש בינארי ניתן למצוא ייחוס עם מינימום תנועה.

כך ב- 30 שניות הראשונות הרובוט ימצא לעצמו נקודת יחוס עם מינימום מאמץ ותנועה ויהיה מוכן לפעולה.

דרך נוספת, אם נוכל למדוד את ממוצע צריכת זרם המנועים ללא תנועה, בנקודת האיזון הצריכה תהייה מינימאלית. כלומר ניתן לשחק עם ההטיה של הרובוט עד צריכה מינימאלית וזאת נקודת האיזון.

להלן הדפסה של זווית הטיה נמדדת לעומת PWM תיקון. מעל 7 מוסיפים 60 לשיפור ביצועי המנוע (שעד 70 לא מגיב כלל). 0.35 מעלה זה סף התגובה כאשר AKp = 20

AKp = 20

-0.20, -4.05

-0.28, -5.52

-0.25, -4.97

-0.32, -6.46

-0.35, -67.03

-0.35, -6.94

-0.49, -69.83

-0.49, -69.71

-0.47, -69.34

-0.44, -68.88

-0.52, -70.32

-0.62, -72.35

-0.68, -73.54

-0.48, -69.52

-0.14, -2.73

-0.05, -0.93

0.02, 0.42

0.01, 0.16

-0.11, -2.29

0.05, 0.94

-0.06, -1.11

-0.09, -1.79

-0.09, -1.86

-0.08, -1.54

-0.03, -0.56

-0.17, -3.39

-0.24, -4.87

-0.32, -6.47

-0.34, -6.76

-0.34, -6.85

-0.42, -68.36

-0.48, -69.65

-0.53, -70.54

-0.58, -71.63

-0.61, -72.19

-0.75, -74.95

-0.76, -75.11

-0.74, -74.88

-0.59, -71.71

-0.35, -67.00

-0.38, -67.65

-0.52, -70.38

-0.44, -68.84

-0.35, -67.06

-0.42, -68.37

-0.45, -68.95

-0.48, -69.54

-0.41, -68.25

-0.23, -4.60

-0.15, -3.01

-0.16, -3.21

-0.16, -3.21

-0.08, -1.55

0.02, 0.38

-0.01, -0.19

-0.01, -0.22

-0.02, -0.49

0.01, 0.30

-0.01, -0.18

-0.08, -1.62

-0.02, -0.36

-0.02, -0.46

-0.09, -1.88

-0.12, -2.49

-0.16, -3.22

-0.24, -4.74

-0.25, -5.09

-0.33, -6.67

-0.42, -68.42

-0.42, -68.46

-0.45, -69.00

-0.38, -67.61

-0.38, -67.55

-0.28, -5.69

-0.15, -2.99

-0.26, -5.29

-0.32, -6.49

-0.36, -67.15

-0.43, -68.62

-0.46, -69.11

-0.47, -69.45

-0.57, -71.48

-0.44, -68.84

-0.28, -5.52

-0.22, -4.32

-0.21, -4.14

-0.20, -3.91

-0.29, -5.76

-0.27, -5.40

-0.28, -5.52

-0.37, -67.33

-0.43, -68.66

-0.51, -70.27

-0.53, -70.61

-0.52, -70.49

-0.56, -71.23

-0.41, -68.20

-0.40, -68.07

-0.42, -68.42

-0.50, -70.07

-0.52, -70.49

-0.41, -68.17

-0.35, -6.93

-0.21, -4.23

0.08, 1.55

-0.12, -2.37

-0.13, -2.64

0.03, 0.68

-0.07, -1.46

-0.03, -0.68

0.05, 0.97

**בעיית ה- Backlash**

הבעיה נחלקת ל- 2

1. Backlash מכני של כ- 2.5 ממ שהם (2.5/(64.5\*pi))\*360 = 4.5 degree (64.5 – קוטר הגלגל ב-מ"מ)
2. Backlash חשמלי של -70 to 70 ב- PWM שהמנוע לא מגיב. ב- KP של 20 מדובר בהטיה של -3.5 to 3.5 = 7 degree ( 70/KP = 3.5 ) בהטיה שהמנוע לא מגיב.

צרוף של 1 ו- 2 מביאם אותנו ל- backlash משוקלל של כ- 11.5 מעלות של חוסר תגובה, הדבר מקשה מאוד בהפסקת התנודות של במערכת.

הגדלת KP ל- 40 מקטינה את ה- backlash החשמלי פי 2 ומעמידה את ה- backlash המשוקלל על כ- 8 מעלות אבל מצד שני מתקבלת תגובת יתר של המערכת הגורמת לתנודות ותזזיתיות וכתוצאה מכך צריכת זרם גבוהה וחימום ה- H-Bridge

כאשר Kp = 20 המערכת רגועה יותר אבל נדרשת הטיה של 255/20 = 12.75 degree להפעלת המנועים בעוצמה מקסימלית, נוסיף לזה backlash מכני של 4.5 מעלות הטיה מגיעים להטיה של כמעט 20 מעלות להפעלה מקסימלית של המנועים. כשהרובוט כולל את הקומה ה- 3 הוא לא מסוגל להתאושש מהטיה זו ומתרסק לרצפה.

לשיפור ה- backlash החשמלי אפשר להחליט על KP לא ליניארי

1. בתיקון של מעל 7 ב- PWM להוסיף עוד 60 והתגובות יחלו כבר לאחר 0.35 מעלות לכל כיוון
2. לקבוע KP של 20 עד מעלה 1, 40 בין 1-2 ו65 בין 2-3 מעלות (יש לכתוב פונקציה השומרת את הרציפות בנקודות ההשקה)

נקבל פונקציה רציפה של KP (כל פונקציה מתחילה היכן שהקודמת מסתיימת) והשיפוע של KP פרופורציוני ל- 0 עבור השורה הראשונה, 20 עבור השנייה, 40 עבור השלישית ו- 65 עבור השורה האחרונה.

במצב רגיל KP הוא קבוע והשגיאה בלבד קובעת את גודל התיקון

כאן KP אינו קבוע וגודלו יקבע על ידי זווית ההטיה

חשוב !!

השגיאה כאן חבויה , כלומר כל ערך השונה מ- 0 הוא שגיאה, כלומר הייחוס תמיד 0

**המשך לבעיית ה- Backlash – הרעידות המכניות משפיעות דרמטית על mpu6050**

הפונקציה להתמודד עם בעיית ה- backlash נועדה להתמודד עם העובדה שמעל זווית הטיה מסוימת (די נמוכה הרובוט לא מאזן עצמו. ההנחה הייתה הבעייתיות של ה- backlash שיוצרת תנודתיות. אבל אז גיליתי שבזויות מסוימות, 10 מעלות בערך, ה- mpu6050 מתחרפן ומיצר מספרים רנדומליים בין 20 ל- 5- מעלות.

ניסיתי לפתור עם קבלים על ה- mpu6050 ועל המנוע ולא עזר. ניתקתי את ה- mpu6050 מ- board (עם חוטים מאריכים) , הקריאה נרגעה. עם המאריכים הדבקתי אותו שוב לboard עם דבק דו צדדי ושוב הקריאה השתוללה. לכן המסקנה המתבקשת רעידות מכניות מהמנוע, שאכן מאוד רועד.

שמתי לב שהקריאה משתוללת רק בעוצמות מסוימות של המנוע, כלומר יש שם איזה תהודה מכנית שמשגעת את ה- MPU

**דרכים לפתרון**

1. לבודד מכנית (בעזרת בולמי זעזועים מתאימים) את המנועים מהרובוט וגם את ה- mpu מה- board
2. לשחק עם תדר הברך של ה- complementary filter, אין לי מושג מי יותר רגיש לרעידות האקסלרומטר או ה- ג'ירו
3. לנסות קלמן פילטר

כמובן שעדיף להתחיל עם הפילטרים.

**הפתרון**

שינוי הברך של ה- complementary filter עד פי 10 למעלה ולמטה , החלפת הפילטר בקלמן פילטר (שנתן בקטנה תוצאות טובות יותר) לא סיפקו תוצאות משמעותיות

לבודד מכנית – בעיה לא פשוטה, המנועים מרעידים הכל!

בא לי הרעיון להוסיף Digital low pass filter על כל אחד מששת הצירים של ה- MPU, מאחר וצריך 6 פילטרים ול- FIR צריך לחשב מקדמים עבור כל תדר וקצב דגימה חשבתי על חלון החלקה המחשב ממוצעים moving average low pass filter כי החישוב פו הוא הקל ביותר ולא משנה גודל החלון

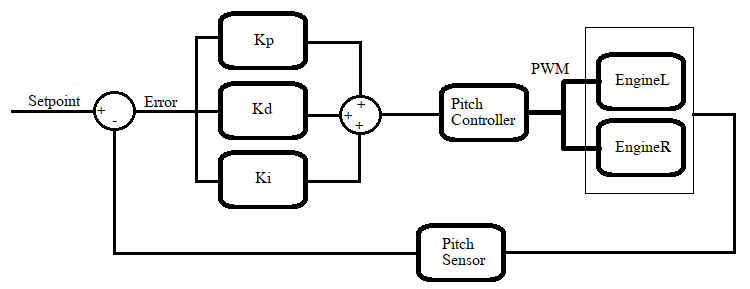
תוך כדי חיפוש ב- MPU-6000-Register-Map1.pdf (מסמך אוגרים ותכנות) מצאתי שיש למערכת DLPF שרק צריך לקנפגו (אוגר 0x1a). חסך ממני את כתיבת הפילטר ובדיקה מהירה הראתה כי נותן להתגבר על רעידות המנוע בעזרת פילטר זה ולקבל קריאות יציבות ב- MPU גם כשהמנועים עובדים בעוצמה מקסימאלית

מתחיל פרויקט חדש

04-BalancingTestWithKalman

שבוא נשתמש ב- DLPF וכן ב- Kalman Filter וגם ב- complementary לשם השוואה

**ה- PID לייצוב הרובוט**

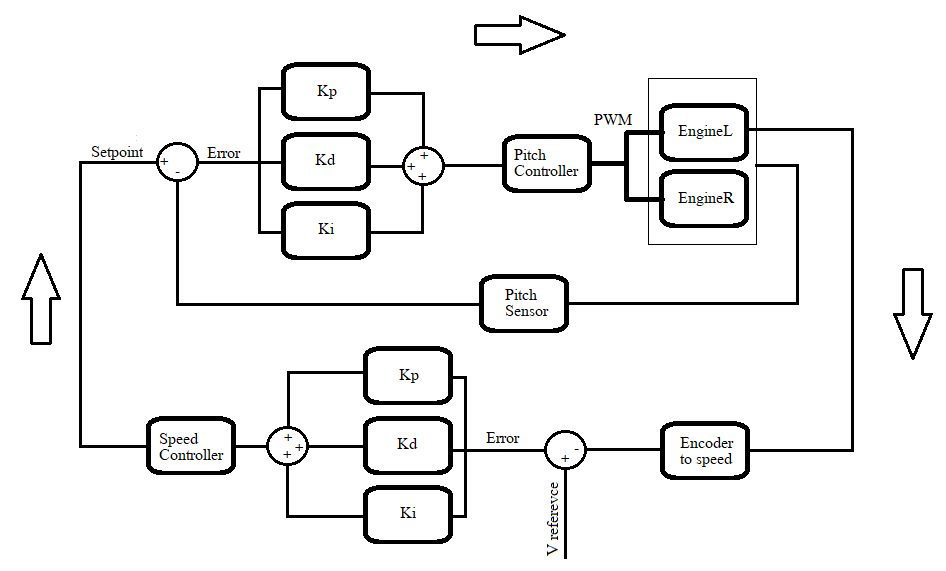


ה- Setpoint שנקבע הוא 0 , כלומר על הרובוט לעמוד זקוף, ברור שהוא יכול לפתח מהיריות אקראיות תוך נסיון הייצוב. בתהליך הייצוב של ה- PID Ki נבחר להיות 0 , כלומר לא משתתף

**תנועה של הרובוט**

הרעיון, שליטה על ה- Setpoint, אם נטה את הרובוט קדימה במעט הוא יחל לנוע בניסיון הייצוב והפוך.

כדי לנוע במהירות קבועה V קדימה יש צורך ראשית למדוד את מהירות הרובוט בעזרת המקודדים של המנוע. להטות את הרובוט קדימה במעט והוא יחל להאיץ. בעזרת חוג סגור יש לשחק עם ההטיה כדי לשמור על המהירות קבועה. לשם כך צריך מערכת PID נוספת



מהאתר <https://github.com/TKJElectronics/KalmanFilter/tree/master> לקחתי העתק של הספרייה ושמרתי ב-

C:\Users\mzeza\_000\Documents\Arduino\libraries\KalmanFilter-master

את הקוד ההתחלתי לקחתי מהדוגמאות הנפתחות ב- IDE תחת ספריה זו

בשלב זה המהירות נלקחה רק מהגלגל השמאלי, גם במערכת ה- PID החדשה Ki הוא 0, כלומר לא בשימוש בשלב זה. זה אומר שכנראה לא מצליחים להגיע ממש למהירות המבוקשת אבל Ki צריך טיפול מיוחד למניעת תנודות. במהירות איטית המנועים לא מגיבים טוב לכן הסכימה מצטברת ואז המנוע נפתח ומתחיל בנסיעה וממשיך גם עם עוצמה נמוכה יותר ולכן לא מאבד מהירות רק כאשר הסכימה הופכת כיוון וכך מתקבלות תנודות ותנועה לא חלקה.

אני לא יכול להעלות את המהירות מאחר והמנוע בעומס מגיע רק לכ 40 ס"מ לשנייה ב- 20 ס"מ לשנייה זה כבר מסוכן לייצוב וב- 10 הוא לא סוחב.

לפתרון בעיה זו צריך מנועים אחרים כגון מנוע צעד או מנוע סרוו הניתנים לשליטה גם במהירות נמוכה

**סיבוב הרובוט**

בגדול ניתן לייצב את הרובוט גם עם גלגל אחד מתוך השניים, אם נשאיר גלגל אחד במקום הגלגל שני יצטרך לעשות פי 2 דרך כדי לייצב את הרובוט. במצב זה הרובוט יעשה "רגל ציר".

לכן אם נשלוט על מהירות כל גלגל בנפרד ובאותו היחס יגדיל את תיקוני הייצוב לגלגל המהיר ונקטין לגלגל האיטי הרובוט יסתובב תוך כדי תנועה וימשיך לייצב עצמו

לשם כך צריך לבקר כל גלגל בנפרד

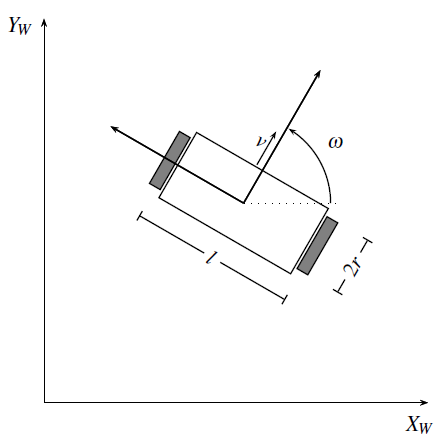
עד עכשיו חישבנו את המהירות הקווית רק על פי גלגל אחד על פי הנוסחה

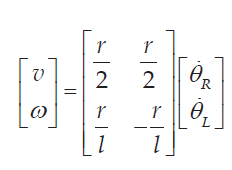
עכשיו שכל גלגל נע במהירות שונה, המהירות של הרובוט היא ממוצע מהירות שכל גלגל תורם

חישבנו ומצאנו כי המהירות הזוויתית של הגלגל היא

נסמן את מהירות הזוויתית של סיבוב של הרובוט ב- גדולה

במאמר מהירות הסיבוב של הרובוט היא ואילו אצלי  *גדולה. כמו כן במאמר מהירות סיבוב הגלגל היא הנגזרת של θ ואצלי ω*





תרגיל:

נניח כי גלגל ימין מסתובב במהירות של 10 ס"מ לשנייה ואילו גלגל שמאל מסתובב במהירות של 5 ס"מ לשנייה

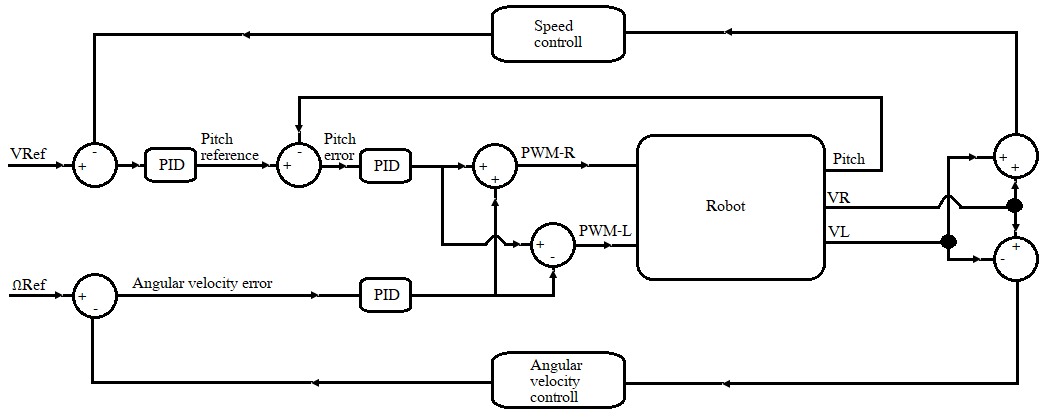
מה צפויה להיות המהירות הזוויתית של הרובוט

*– המרחק בין הגלגלים הוא 18.5 ס"מ*

*נציב בנוסחה ונקבל*

*נציב את המספרים ונקבל*

**מבנה המערכת**



ספריית PID בארדווינו

<http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>

<https://www.youtube.com/watch?v=I6z26LVu5y0>

<https://maker.pro/arduino/projects/build-arduino-self-balancing-robot>

<https://pdfs.semanticscholar.org/1672/c4c03c2c59fbbcad12797675d5a0a8f37525.pdf>

<https://www.youtube.com/watch?v=6oYTeo6ZIak>

<http://rztronics.com/build-self-balancing-robot-using-arduino-smartphone-control/>

<https://www.youtube.com/watch?v=poWP-RpPa3g>