

### תרגיל בית תיאורטי 5 – בקרת PID- קבוצה 39

318650918	205919343	314789165	316136084
-----------	-----------	-----------	-----------

1. א.

$$\sum M = I\ddot{\theta}$$

$$I\ddot{\theta} = mgl \sin \theta - c \cdot \dot{\theta} + \tau$$

נבצע לינאריזציה שכן בעבור זוויות  $|\theta| < 15$  זוויות  $\sin \theta \approx \theta$

$$I\ddot{\theta} = mgl\theta - c \cdot \dot{\theta} + \tau$$

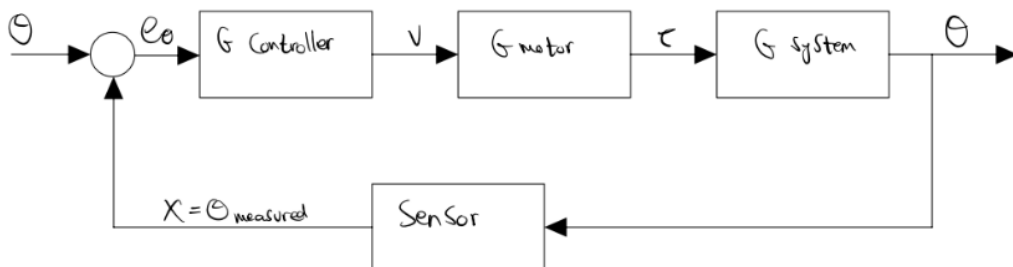
$$\ddot{\theta} = \frac{mgl}{I} - \frac{c}{I}\dot{\theta} + \frac{\tau}{I}$$

$$\dot{x} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \frac{mgl}{I} & -\frac{c}{I} \end{pmatrix} * \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{I} \end{pmatrix} \cdot \tau$$

$$= \begin{pmatrix} 0 & 1 \\ 100 & -22.222 \end{pmatrix} * \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 7407.41 \end{pmatrix} * \tau$$

$$Y = (1 \ 0) * \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} = \theta$$

ב. דיאגרמת הבקרה של החוג הסגור



2. א. להלן פונקציית התמסורת

$$H_{Sensor} = \frac{X}{\theta_{measured}}$$

$$G_{motor} = k = \frac{1}{s}$$

$$G_{controller} = \frac{v}{e(\theta)} = k_p + k_d \cdot s + \frac{k_i}{s}$$

על מנת למצוא את פונקציית התמסורת של  $G_{system}$  נשתמש במודל שמצאנו ונבצע התמרת לפלס.  
תנאי התחלה

$$\theta(0) = 0, \dot{\theta}(0) = 0$$

$$I\ddot{\theta} = mgl\theta - c \cdot \dot{\theta} + \tau$$

$$L(I\ddot{\theta}) = I * (s^2 * \theta(s) - s * \theta(0) - \dot{\theta}(0)) = I * s^2 * \theta(s)$$

$$L(\tau + mgl\theta - c \cdot \dot{\theta}) = mgl\theta(s) - c \cdot \theta(s) \cdot s + \tau$$

$$I * s^2 * \theta(s) = mgl\theta(s) - c \cdot \theta(s) \cdot s + \tau$$

$$\tau = \theta(s)(I * s^2 - mgl + cs)$$

$$G_{system} = \frac{\theta(s)}{\tau} = \frac{1}{I * s^2 - mgl + cs}$$

ב. פונקציית התמסורת של כלל המערכת בחוג סגור

$$\begin{aligned} G_{Total} &= \frac{G_{controller}(s) * G_{motor} * G_{system}(s)}{1 + G_{controller}(s) * G_{motor} * G_{system}(s) * sensor} = \\ &= \frac{\frac{1}{5} * \left(k_p + k_d \cdot s + \frac{k_i}{s}\right) * \frac{1}{I * s^2 + cs - mgl}}{1 + k_p * \frac{1}{5} * \frac{1}{I * s^2 + cs - mgl}} = \\ &= \frac{0.2 * \left(k_p + k_d \cdot s + \frac{k_i}{s}\right)}{(I * s^2 + cs - mgl) + 0.2 \left(k_p + k_d \cdot s + \frac{k_i}{s}\right)} = \\ &= \frac{k_p + k_d \cdot s + \frac{k_i}{s}}{5(I * s^2 + cs - mgl) + k_p + k_d \cdot s + \frac{k_i}{s}} \end{aligned}$$

ג. פרמטר הבקרה המתאים הינו

$$p.o = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \leq 0.2 \rightarrow \frac{\zeta\pi}{\sqrt{1-\zeta^2}} \leq 1.609 \rightarrow \zeta^2 \leq 0.262 * (1 - \zeta^2) \rightarrow$$

$$\zeta = 0.4556 < 1$$

ולכן אנו נמצאים בתת ריסון. מתוך הנחה שהקלט הוא כניסת מדרגה בגודל יחידה אחת, נקבל כי שגיאת המצב המתמיד הינה:

$$X_d(s) = \frac{1}{s} \rightarrow X(s) = G_{Total} * X_d(s) = \frac{1}{s} * \frac{k_p * \frac{1}{5 * I}}{s^2 + \frac{cs}{I} + \frac{k_p}{I} - \frac{mgl}{I}}$$

$$G_{Total} = \frac{k_p * \frac{1}{5 * I}}{s^2 + \frac{cs}{I} + \frac{k_p}{5 * I} - \frac{mgl}{I}} = \frac{k_p * \frac{1}{5 * I}}{s^2 + 2\omega_0\zeta * s - \omega_0^2}$$

$$\omega_0 = \frac{c}{2\zeta I} = 24.38$$

$$\omega_0^2 = \frac{k_p * \frac{1}{5} - mgl}{I} = 594.666 \rightarrow k_p = 0.468$$

$$X(t) = \lim_{s \rightarrow 0} sX(s) = \lim_{s \rightarrow 0} \frac{s * \frac{1}{s} * k_p * \frac{1}{5 * I}}{s^2 + \frac{cs}{I} + \frac{k_p}{5 * I} - \frac{mgl}{I}} = \frac{k_p}{k_p - 5 * mgl} =$$

$$\frac{0.468}{0.468 - 5 * 13.5 * 10^{-3}} = 1.168$$

נשווה מקדמים ונקבל  $\omega_0 = \frac{c}{2\zeta I} = 24.37$  ולכן שגיאת המצב המתמיד הינה:

$$\lim_{t \rightarrow \infty} X(t) = 1 - 1.168 = -0.1685$$

3. א.

$$G_{controller}(s) = K_p + K_d * s \quad \text{פונקציית התמסורת של הבקר}$$

$$sensor = 1 \quad \text{פונקציית התמסורת של החיישן}$$

$$G_{motor} = k = \frac{1}{5} \quad \text{פונקציית התמסורת של המנוע}$$

$$G_{system}(s) = \frac{1}{Is^2 + cs - mgl} \quad \text{פונקציית התמסורת של המערכת}$$

ב. נמצא את פונקציית התמסורת של המערכת בחוג סגור

$$G_{total}(s) = \frac{G_{controller}(s) * G_{motor} * G_{system}(s)}{1 + G_{controller}(s) * G_{motor} * G_{system}(s) * sensor}$$

$$= \frac{(K_p + K_d * s) * \frac{1}{5} * \frac{1}{I * s^2 + c * s - mgl}}{1 + (K_p + K_d * s) * \frac{1}{5} * \frac{1}{I * s^2 + c * s - mgl} * 1}$$

$$= \frac{(K_p + K_d * s) * \frac{1}{5}}{(I * s^2 + c * s - mgl) * (1 + (K_p + K_d * s) * \frac{1}{5} * \frac{1}{I * s^2 + c * s - mgl} * 1)}$$

$$= \frac{(K_p + K_d * s) * \frac{1}{5 * I}}{I * s^2 + c * s - mgl + (K_p + K_d * s) * \frac{1}{5}} = \frac{(K_p + K_d * s) * \frac{1}{5 * I}}{s^2 + \left(\frac{c}{I} + \frac{K_d}{5 * I}\right) * s + \frac{K_p}{5 * I} - \frac{mgl}{I}}$$

ג. פרמטרי הבקרה המתאימים עבור  $overshoot \leq 20\%$  ו  $|e_{ss,pd}| \leq 0.5 * |e_{ss,p}|$

הינם:

$$overshoot \leq 20\% \rightarrow P.O = e^{-\frac{\zeta * \pi}{\sqrt{1-\zeta^2}}} \leq 0.2 \rightarrow \zeta = 0.4556$$

כיוון שהתקבל  $\zeta < 1$  אנו נמצאים בתת ריסון.

כעת נרצה למצוא את  $K_p$  כתלות ב  $\omega_0$ :

$$|e_{ss,pd}| \leq 0.5 * |e_{ss,p}| \rightarrow \frac{1}{1 + \frac{K_p}{\omega_0^2}} \leq 0.5 * 0.1685 = 0.08425 \rightarrow K_p = 10.87 * \omega_0^2$$

מתוך ההנחה שהקלט הינו כניסת מדרגה בגודל יחידה אחת,  $X_d(s) = \frac{1}{s}$  נקבל:

$$X(s) = G_{total}(s) * X_d(s) = \frac{(K_p + K_d * s) * \frac{1}{5 * I}}{s^2 + \left(\frac{c}{I} + \frac{K_d}{5 * I}\right) * s + \frac{K_p}{5 * I} - \frac{mgl}{I}} * \frac{1}{s}$$

נשווה מקדמים ונמצא את פרמטרי הבקרה

$$G_{total}(s) = \frac{(K_p + K_d * s) * \frac{1}{5 * I}}{s^2 + \left(\frac{c}{I} + \frac{K_d}{5 * I}\right) * s + \frac{K_p}{5 * I} - \frac{mgl}{I}} = \frac{1}{s^2 + 2 * \omega_0 * \zeta * s + \omega_0^2}$$

$$\omega_0^2 = \frac{K_p * \frac{1}{5} - mgl}{I} = \frac{2.1738 * \omega_0^2 - 13.5 * 10^{-3}}{1.35 * 10^{-4}} \rightarrow 2.174 * \omega_0^2 = 13.5 * 10^{-3} \rightarrow$$

$$\omega_0 = 0.0788, \quad K_p = 0.0675$$

$$2 * \omega_0 * \zeta = \left(\frac{c}{I} + \frac{K_d}{5 * I}\right) \rightarrow \omega_0 = \frac{5 * c + K_d}{10 * \zeta * I} = 0.0788 \rightarrow$$

$$K_d = 4.85 * 10^{-5} - 5 * 3 * 10^{-3} = -0.015$$

ממשפט הערך הסופי

$$\begin{aligned} \lim_{t \rightarrow \infty} X(t) &= \lim_{s \rightarrow 0} s * X(s) = \lim_{s \rightarrow 0} s * \frac{(K_p + K_d * s) * \frac{1}{5 * I}}{s^2 + \left(\frac{c}{I} + \frac{K_d}{5 * I}\right) * s + \frac{K_p}{5 * I} - \frac{mgl}{I}} * \frac{1}{s} \\ &= \lim_{s \rightarrow 0} \frac{\frac{K_p}{5 * I}}{\frac{K_p}{5 * I} - \frac{mgl}{5 * I}} = \frac{K_p}{K_p - 5mgl} \end{aligned}$$

לאחר הצבת הפרמטרים נקבל כי שגיאת המצב המתמיד הינה

$$\begin{aligned} e_{ss,pd} &= 1 - \lim_{t \rightarrow \infty} X(t) = 1 - \frac{K_p}{K_p - 5mgl} = 1 - \frac{0.0675}{0.0675 - 5 * 13.5 * 10^{-3}} \\ &= 19,927.964 \end{aligned}$$

### תרגיל בית מעשי בקרת PID – קבוצה 39

#### מטרת הפרויקט:

מטרת המערכת הינה לייצב את מוט המטוטלת בזווית רצויה בטווח של  $\pm 15^\circ$  באמצעות מערכת מבוקרת. המערכת מזהה שגיאה כהפרש המעלות בין הזווית הרצויה למצויה ובכך מתרגמת אותה לכוח שעל המנוע להפעיל בכדי להביא למזעור השגיאה.

#### הנחות יסוד בפיתוח המערכת:

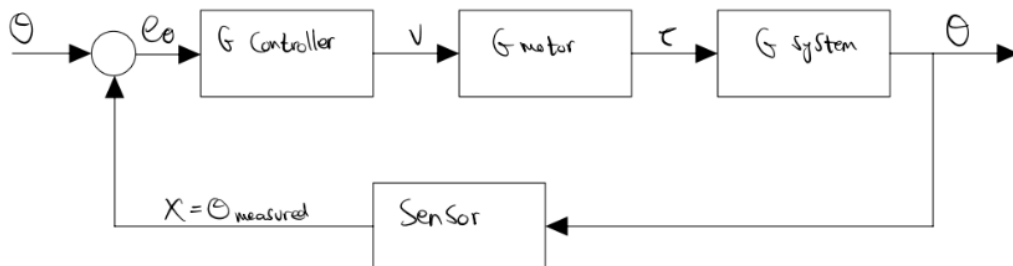
- הזווית תוסט כשהמטוטלת פונה כלפי מעלה.
- לאחר איפוס, המטוטלת מוסטת ממקומה לזווית של עד  $\pm 45^\circ$  ומשחוררת.
- המשתמש יזין קלט שהינו הזווית שאליה ירצה שהמערכת תתייצב בתחום של  $\pm 15^\circ$ .

#### תיאור מצבי קיצון ושיטת הפתרון:

הפרעות חיצוניות – כאשר כוחות חיצוניים או הפרעות פועלים על הזרוע, היא עשויה לסטות מהזווית הרצויה. בקר PID מתאים באופן רציף את ערך PWM של המנוע כדי לנטרל את ההפרעות ולהחזיר את הזרוע לזווית הרצויה. זה מבטיח שהאובייקט יישאר יציב בזווית המטרה, גם בנוכחות כוחות חיצוניים.

#### תיאור התוכן:

הבקר מקבל כקלט את האות של השגיאה אשר חישבנו כהפרש בין ערך הזווית הרצויה (קלט המשתמש) לבין הערך המתקבל בפועל. לאחר התיקון המבוסס על שלושה רכיבים ( $K_i, K_p, K_d$ ) ופרמטר תצא פקודה למנוע.



## טבלה 1 – תיאור הפינים של הבקר :

מספר הפין	שם המשתנה	תצורה	הפעלה
11	MOTOR_CCW	Output	בהינתן שגיאה הבקר ידע לשלוח את כיוון הסיבוב הרצוי ל-H-Bridge דרך אחד מהפינים בהתאמה
10	MOTOR_CW	Output	
9	enA	Output	מקבל את גודל הכוח שהמנוע צריך להפעיל ושולח אותו
2	Encoder_A	Input	הבקר מקבל כקלט את ערך וסימן השגיאה ובאמצעות ספריית PID מחשב פקודה למנוע
3	Encoder_b	Input	

## תיאור תהליך הכיול:

במהלך התרגיל התיאורטי חישבנו את הפרמטרים עבור מטוטלת בעלת שלוש משקולות ונתונים זהים. לאחר שהרכבנו את המערכת וכתבנו את הקוד המתאים, ביצענו התאמות בין ערכי הקבועים שקיבלנו לבין המערכת הפיזית בפועל. בכדי לבצע התאמה בין שני ערכים אלו, חישבנו בשנית את הפרמטרים עבור מטוטלת בעלת שתי משקולות. ביצענו הזנת קבועים בקוד המערכת ואחר כך ערכנו התאמות בכדי להבין את ההשפעה של ערכים אלו על יציבות המערכת.

בשלב הראשון בחרנו  $K_i$  רנדומלי וראינו כי אין מספיק כוח למערכת בכדי להזיז את המטוטלת. כתוצאה מכך, בחרנו להגדיל את ה- $K_p$  שאכן הגביר את תגובת המערכת כמצופה.

בשלב השני ראינו כי הגדלת ה- $K_p$  יצרה תגובת יתר, ולכן ריסנו זאת על ידי הקטנת ה- $K_d$ .

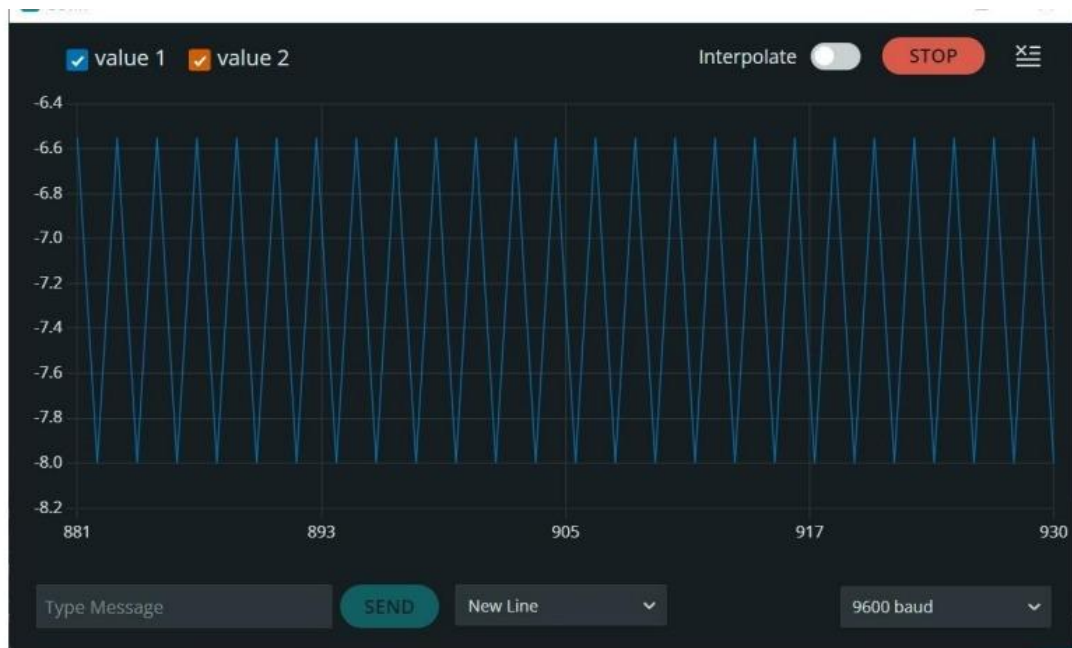
	בקר P (תרגיל תיאורטי)	בקר PD (תרגיל תיאורטי)	בקר PID (תרגיל מעשי)
$K_p$	0.468	0.0675	10
$K_i$	0	0	1.7
$K_d$	0	-0.015	4.1

## תיאור גרפי בקר PID

הפולס הראשוני לאחר הכנסת הזווית הרצויה כקלט

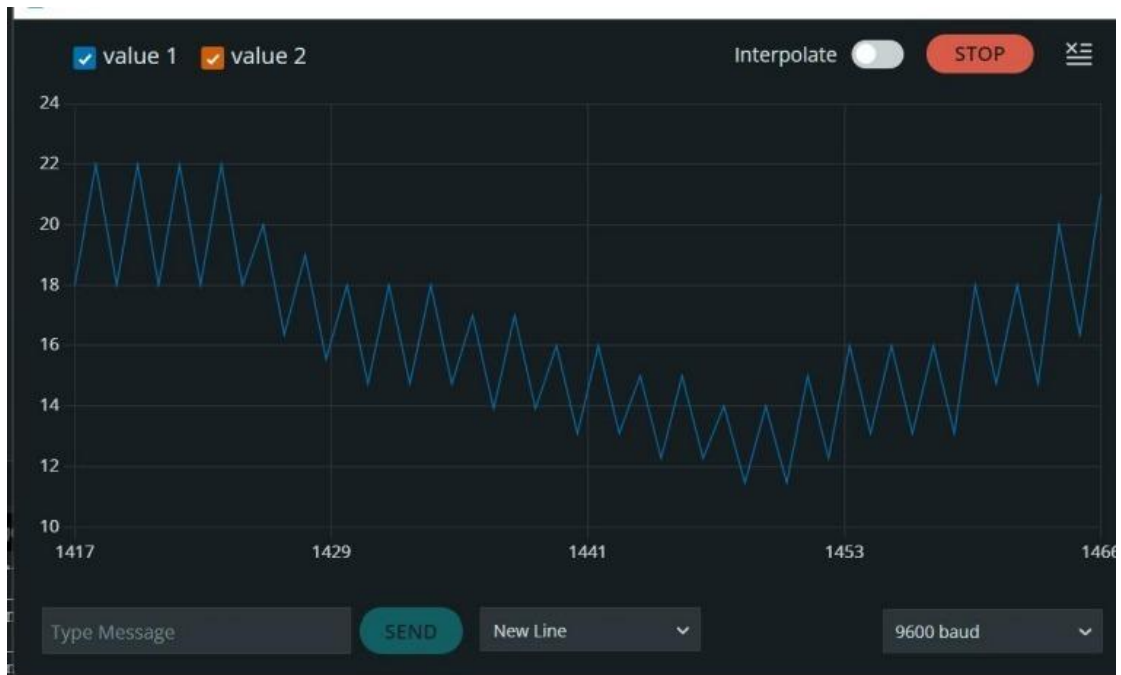


המערכת באמצע הריצה

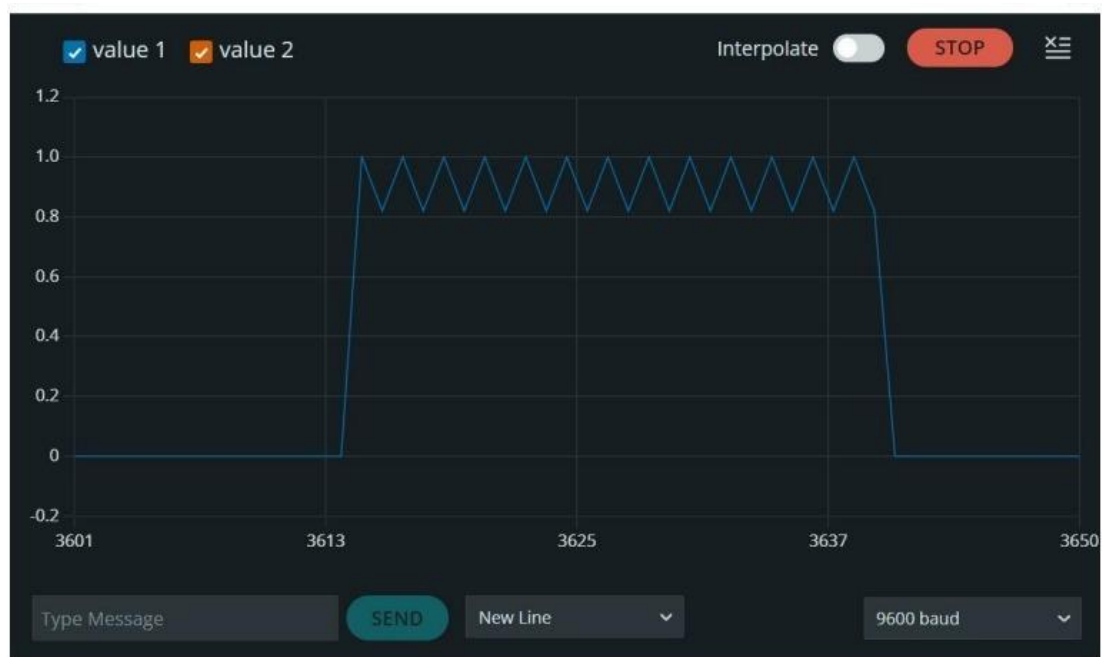




## המערכת לקראת התייצבות



## המערכת מתייצבת



## **מסקנות והמלצות:**

במהלך העבודה בכדי למצוא את ערכי הפרמטרים המתאים שיגרמו לאיזון וייצוב המערכת כמצופה, ביצענו ניסוי וטעיה. גילינו כי קיים פער משמעותי בין הערכים שחושבו בתרגיל התיאורטי לאלה שבפועל גרמו לייצוב המערכת.

הסיבות האפשריות לפערים אלו הן:

- שונות במנועים
- אורך מוט המטוטלת ומיקום המשקולות על המטוטלת בפועל
- הפרשים במשקלי המשקולות על המטוטלת
- לאחר בדיקת משקל המשקולות, גילינו כי קיים הפרש בין המשקל הנתון בתרגיל התיאורטי ולבין המשקל בפועל. הפרש זה ככל הנראה הביא לשוני בין ערכי הפרמטרים.
- המומנט שחושב בתרגיל התיאורטי מתייחס למשקל נקודתי ולא תואם למציאות, שכן בפועל למשקולת גודל וצורה וכן המשקולת אינה גוף נקודתי.

לסיכום, בכדי ליצור מערכת אמינה בעלת רגישות נמוכה יותר לשינויים בין המודל השונים, נמליץ על שימוש ברכיבים חזקים ומדויקים יותר.

```

#include <Encoder.h>
#include <PIDController.h>
#define MOTOR_CCW 11
#define MOTOR_CW 10
#define enA 9
#define ENCODER_A 2
#define ENCODER_B 3 // Encoder output to Arduino Interrupt pin ncoderPinB
int motorPwmvalue;
int PowerInPercentage; //power of dc motor

PIDController pid;

int encoder_count = 0; // stores the current encoder count
float changeToDegree1;
double Kd=4.1;
double Ki=1.7;
double Kp=10;
float goal_deg=0;

Encoder myEnc(ENCODER_B, ENCODER_A);

*/setup/**
void setup() {
  Serial.begin(9600)
  pinMode(enA, OUTPUT);
  pinMode(MOTOR_CW, OUTPUT);
  pinMode(MOTOR_CCW, OUTPUT);

  // Set initial rotation direction

```

```

digitalWrite(MOTOR_CW, HIGH);
digitalWrite(MOTOR_CCW, LOW);

pid.begin(); // Initialize the PID controller
pid.limit(-255, 255); // The PID output can be limited to values between -255
to 255
pid.tune(Kp, Ki, Kd); // Set PID parameters
pid.setpoint(goal_deg); // the goal angle
Serial.print("Please choose the goal degree you desire:");

{
void loop} ()
    if (Serial.available()>0){
        get_user_input;()
    {
        encoder_count = myEnc.read;()

        changeToDegree1 = changeToDeg(encoder_count); //change the encoder
to degree

        while (abs(changeToDegree1)>360){ // change back to range 0-360
            if (changeToDegree1<0)
                changeToDegree1=changeToDegree1+360;
            else
                changeToDegree1=changeToDegree1-360    ;
        }
// prints the current deg and the goal deg
Serial.print("changeToDegree1 = ");
Serial.println(changeToDegree1);
Serial.print("The goal degree is: ");
Serial.println(goal_deg);
delay;(1)

```

```

    motorPwmvalue = float(pid.compute(changeToDegree1)) ;

// changes the direction due to the erroe
if (motorPwmvalue <0 )
    reverse;()
else
    forward ;()

// send PWM value to motor
analogWrite(enA, abs(motorPwmvalue));

delay;(10)
{

*/functions/**
float changeToDeg(long tick){
    return tick*360.0/440.0;
}

*/functions/**
void forward}(){
    digitalWrite(MOTOR_CW, HIGH);
    digitalWrite(MOTOR_CCW, LOW);
}

void reverse}(){
    digitalWrite(MOTOR_CW, LOW);
    digitalWrite(MOTOR_CCW, HIGH);
}

```

```
{

void get_user_input(){
    goal_deg= Serial.parseFloat();
    if (abs(goal_deg)> 15){
        Serial.print("not valid degree");
        goal_deg = 0;
    }
    Serial.read(); // get the ENTER char from the serial
}
```

### **הוראות למפעיל**

- בכדי להשתמש במערכת יש לחבר את הבקר למחשב ע"י חיבור USB.
- לאחר מכן יש לרסות את המערכת, פעולה זו מתבצעת על ידי לחיצה על הכפתור האדום והחזקת המטוטלת ב-0 מעלות.
- לאחר הריסות על המשתמש להזין קלט של זווית בין  $15 < X < 15$  - מעלות והתוכנית תתחיל לרוץ.
- בעת הריצה המטוטלת תתאפס על המעלות שהוזנו בקלט.