

## מטרת הפרויקט

מטרת המערכת שבנינו ע"י ארדואינו PID, היא לייצב את מוט המטוטלת ההפוכה בזווית המוגדרת ע"י המשתמש בתחום זוויות של  $\pm 15^\circ$  תוך שימוש בקרת PID על המערכת. משימת הבקרה הינה לחשב את הקבועים  $K_d, K_p$  על סמך המודל התאורטי על מנת לקבל אינדיקציה מה המדדים שצריך לקבוע.

## הנחות יסוד

1. המשתמש יכניס אך ורק זוויות למערכת.
2. לאתחול המערכת נדרשת דחיפה קטנה ידנית למטוטלת.
3. הזוויות שהמשתמש יכניס יהיו רק בין 0-360 מעלות.

## מצבי קיצון ושיטות פתרון

אחרי חשיבה מרובה לגבי סעיף זה הגענו למסקנה כי בקוד לא התייחסנו למצבי קיצון ספציפיים הדבר נובע מכך כי כל הפונקציות גנריות ורובן מועתקות מהתרגול עם ההתאמות הנדרשות לתרגיל הנ"ל.

## תיאור התכן

טבלת משתנים

מספר הפין	שם המשתנה	תצורה	הפעלה
2	ENCODER_A	INPUT	קלטים המשמשים לקריאת האותות מהמקודד
3	ENCODER_B	INPUT	
9	enA	OUTPUT	פלט למהירות סיבוב המנוע
10	MOTOR_CW	OUTPUT	מגדיר את כיוון סיבוב המנוע
11	MOTOR_CCW	OUTPUT	

## תיאור תהליך הכיול

חישוב המשתנים מהחלק התאורטי:

הפרמטר וערכו	ערך הפרמטר הסופי
$K_p > 0.8723$	$K_p = 8.7$
$K_d = 0.0062$	$K_d = 0.62$

$Ki = 2$	$Ki = 3$
----------	----------

**הפרמטר  $k_p$**  - מייצג את החלק של התגובה הפרופורציונלית של הבקר לשגיאה. קיבלנו ערך מאוד מאוד קטן ל $k_p$  מה שהקשה על תגובה של המערכת ולכן היינו צריכים להגדיל את ערכו כדי שהבקר יגיב חזק. הפלט של הבקר ישתנה בצורה מובהקת יותר ככל שהשגיאה תגדל. ערך נמוך של  $k_p$  יוביל לתגובה נמוכה לשגיאה, כלומר הפלט של הבקר ישתנה רק מעט עם שינוי השגיאה ולכן הערך הנמוך תחילה לא הספיק. המערכת הצליחה להתייצב כשהגדלנו פי 10 את  $k_p$ .

**הפרמטר  $k_i$**  - מייצג את החלק האינטגרלי של הבקר. סוכם את שגיאות העבר לאורך זמן ועוזר לבטל את השגיאה השירית כאשר המערכת נמצאת במצב יציב, למשל אם יש הטיה במערכת. ערך גבוה של  $k_i$  כמו ששמנו מתקן את הטעות המתמשכת ולכן בחרנו בערך יחסית גבוהה לקבלת מערכת שלמה יותר.

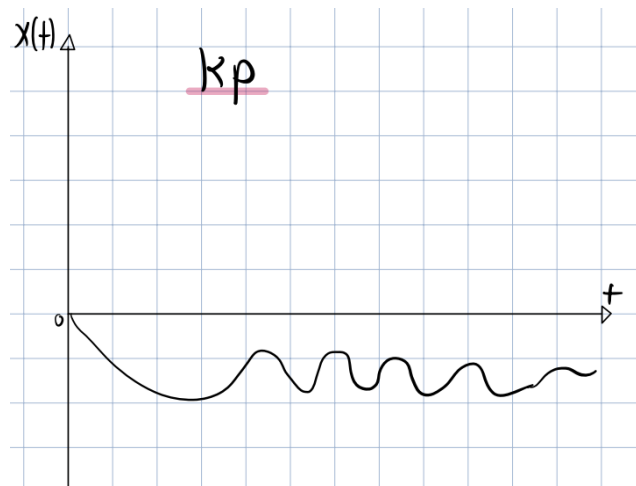
**הפרמטר  $k_d$**  - מייצג את החלק הדיפרנציאלי של הבקר, תפקידו העיקרי הוא להפחית את החריגות והתנודות במערכת הבקרה שנוצרים ע"י  $k_i$ . ערך גבוה של  $k_d$  פירושו שהבקר יגיב הרבה לשינויים בשגיאה התנודד פחות, וערך נמוך פירושו שהבקר יגיב מעט לשינויים בשגיאה. המערכת הצליחה להתייצב רק כשהגדלנו פי 100 את ערך  $k_d$ .

## השוואה בין ערכי הפרמטרים

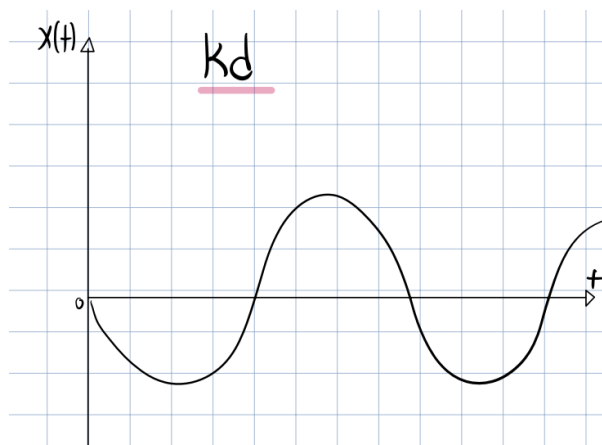
כשפתרנו את התרגיל התאורטי קיבלנו לכל משתנה ערכים. למשתנה  $k_p$  קיבלנו טווח ערכים הגדול מ-0.87 וערך ספציפי ל $k_d$ . לאחר בדיקה עם הבקר עצמו גילינו כי הערכים האלה נמוכים מידי על מנת לאפשר לבקר להגיב בצורה טובה ולכן הקבועים שהצבנו בתרגיל המעשי היו גדולים אך עומדים בתנאים שקיבלנו בערך התאורטי.

### גרפים עבור בקר $P$ ובקר $PD$

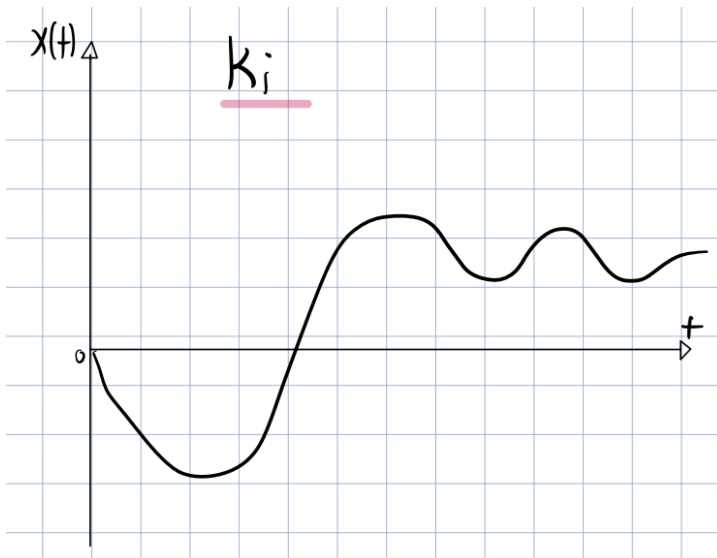
$K_P$  שלנו בהתחלה היה מאוד נמוך מה שהקשה על המטוטלת לזוז ולהגיע לזווית הרצויה, הגרף היה נראה כך:



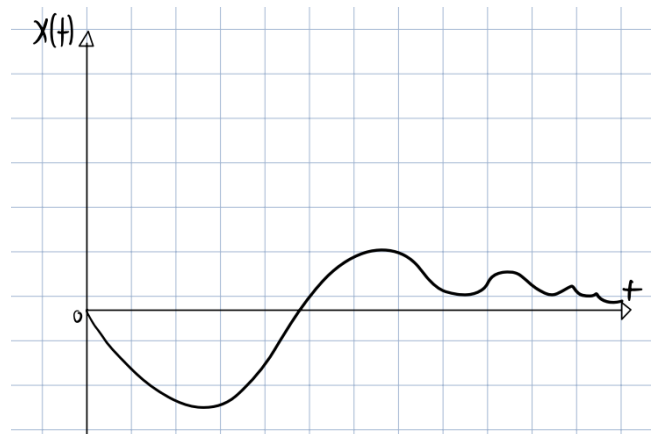
$K_d$  שלנו גם היה מאוד מאוד קטן ולכן הייתה סטייה גדולה מאוד של המטוטלת מנערך הרצוי, הגרף נראה כך:



כשהגדרנו את  $K_i$  הצלחנו לשפר את הסטייה של המערכת ולתקן את השגיאות שלה , הגרף נראה כך:



לבסוף המערכת שלנו נראתה כך:



## תיעוד קוד התוכנית

```
define ENCODER_A 2#
define ENCODER_B 3 // Encoder output to Arduino Interrupt pin ncoderPinB#
define MOTOR_CCW 11#
define MOTOR_CW 10#
define enA 9#
#include <Encoder.h#
#include <PIDController.h#

float encoder_count; // stores the current encoder count
;float tick_to_deg1
;int motor_pwm_value
;int userInput

;PIDController pid
;Encoder myEnc(ENCODER_B, ENCODER_A)

/***** functions*****/
void forward(){ // move motor FORWARD
```

```

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

void reverse(){ //move motor BACKWARDS
;digitalWrite(MOTOR_CW, LOW)
;digitalWrite(MOTOR_CCW, HIGH)
{

}()void get_user_input
;()userInput = Serial.parseFloat
Serial.read(); // get the ENTER char from the serial
;Serial.println(userInput)
{

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

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

pid.begin(); // Initialize the PID controller
pid.limit(-255, 255); // The PID output can be limited to values between -255 to 255

```

```

pid.tune(8.7,2,0.62); // Set PID parameters

;Serial.print("Write the desired angle: ")
{

}

()void loop
if (Serial.available()>0){ // Wait for User to Input Data
;()get_user_input
{
;()encoder_count = myEnc.read
;tick_to_deg1 = tick_to_deg(encoder_count)
}while (abs(tick_to_deg1)>360)
if (tick_to_deg1<0)
;tick_to_deg1=tick_to_deg1+360
else
;tick_to_deg1=tick_to_deg1-360
{

motor_pwm_value = float(pid.compute(tick_to_deg1)); //compute the PWM value for
the motor based on current angle

calculate motor direction //
}if (tick_to_deg1 <userInput )
;()reverse
{
} else
;()forward
{

send PWM value to motor //

```

```
;analogWrite(enA, abs(motor_pwm_value))  
;(1)delay  
{
```

## סיכום ומסקנות

בקרת PID של מוטטלת מטרתו לשלוט בזווית של מוטטלת באמצעות בקר PID. מטרת הפרויקט היא להשתמש בבקר PID על מנת לייצב מוטטלת ע"י הכנסת זווית רצויה. הפרויקט כלל מספר שלבים

1. שלב תאורטי שבו היינו צריכים למדל את המערכת ולמצוא את הפרמטרים הרצויים על מנת להיעזר בהם בחלק המעשי.
2. מידול המערכת של המוטטלת ע"י בניית המערכת פיזית וכתובת קוד מתאים והבנת הקשר בין הקלט לפלט.
3. בחירת ערכים מתאימים על מנת לכוון את הבקר על פי ניסיונות רבים והבנת המערכת ומה ההתנהגות הרצויה.

## נספח

### הנחיות מפעיל

1. יש לחבר את המערכת לחשמל ולמחשב עם קוד הארדואינו.
2. יש להזין זווית רצויה באמצעות ממשק המשתמש ב serial monitor .
3. יש להזיז את המוטטלת ולחכות להתייצבות המערכת.