**דו"ח Aero Simulation – מעבדה מתקדמת למערכות אוטונומיות**

**רקע תיאורטי:**

ראשית, נסקור את בקר הPID – בבקר קיימים שלושה מרכיבים מרכזיים – רכיב הגבר, רכיב אינטגרציה ורכיב גזירה. רכיב ההגבר מגביר את אות השגיאה בין המיקום הנוכחי והרצוי, אך gain גבוה יכול להוביל לחריגה ולתנודתיות. רכיב הגזירה מתמקד בקצב השינוי במיקום כדי לדכא תנודות. רכיב האינטרציה, המבוסס על השגיאה המצטברת לאורך זמן, מתקן טעויות במצב יציב. בעבודה נעסוק רבות בבקרי PID – הן במובן התיאורטי והן בעיסוק המעשי בסימולציית הAERO.

כעת נסקור את מערכת הAERO – בחלק זה, עלינו לשלוט בגובה ובמיקום האופקי של הרחפן המדומה, Quanser Aero VTOL. הבקרה כוללת התאמת הגובה והדחף של ה-Quanser Aero, עם פתרונות של בקרת PID מקוננת פשוטה. הרחפן מתחיל על הקרקע ועליו לנווט דרך נקודות ציון, תוך שמירה על מיקום בגבולות השגיאה שצוינו. האתגרים כוללים שינוי בתנאי הרוח שמפריעים לבקר והוספת מורכבויות נוספות למשימה.

\*דיאגרמת בלוקים של המערכת (PID ובפרט של הAERO) תופיע בחלק השני בשאלות התיאורטיות.

לראות אם צריך לעשות יחידות זמן לגרפים+ LEGEND – לקבל תשובה מולדימיר

Theoretical background Questions

1. נציע מתודה לזיהוי פונקציית התמסורת של מערכת לינארית מתוך תגובת המדרגה שלה:

כידוע, ניתן לבטא את פונקציית המדרגה במישור התדר באופן הבא - . כמו כן, מהגדרת התמסורת: , כאשר במקרה שלנו היא פונקציית מדרגה נקבל שפונקציית התמסורת של המערכת הליניארית *הינה .*

1. תמסורת מסדר שני, הינה מהצורה:

*כמו כן, אנו יודעים את הקשר בין המקדמים לבין קשרים שונים במערכת, בפרט:*

* OS (OverShoot) = *(תגובת היתר)*
* Rising Time = *(זמן העלייה אשר מחושב כזמן בין הערך הסופי ב10% לבין הערך הסופי ב90%)*
* *הערך שעליו נתייצב*

*את כל הערכים הללו נוכל לחשב ולמצוא מהגרף ועל כן באמצעות שתי המשוואות הראשונות נוכל למצוא את: וע"פ הערך הסופי היציב את* K *ובכך מצאנו את התמסורת הרצויה.*

1. בהינתן התמסורת הבאה:

נרצה לממש בקר PID אשר עבור תגובת מדרגה יספק:

* Overshoot=0
* Steady-State-Error = 0
* Settling time < 0.25 seconds

לפי הנוסחאות שהראינו לעיל, ע"מ למצוא Overshoot = 0 נצטרך להשתמש ב זהותית. כמו כן, ע"מ למצוא את ה אנו ניעזר בהגדרות של ושל ההגבר. בגלל שהביטוי *במכנה מורכב עם 10 אזי וכן לפי הנוסחה לזמן ההתייצבות:*

כלומר לא עמדנו בדרישות שכן אם דרשנו עבור הOS אזי לא נוכל להגיע לזמן ההתייצבות שרצינו לפי הנוסחה. כמו כן, נציין כי כדי להגיע לשגיאת מצב מתמיד 0 עלינו להוסיף אינטגרטור לבקר שלנו שכן ע"פ טבלת מצבי השגיאות המתמידות נצטרך מערכת type-1. לסיכום, הדרישות שלא יכלו להיות ממוצות הינן הOS וזמן ההתייצבות שכן ניתן לספק אחת מהן אך לא את שתיהן כפי שראינו.

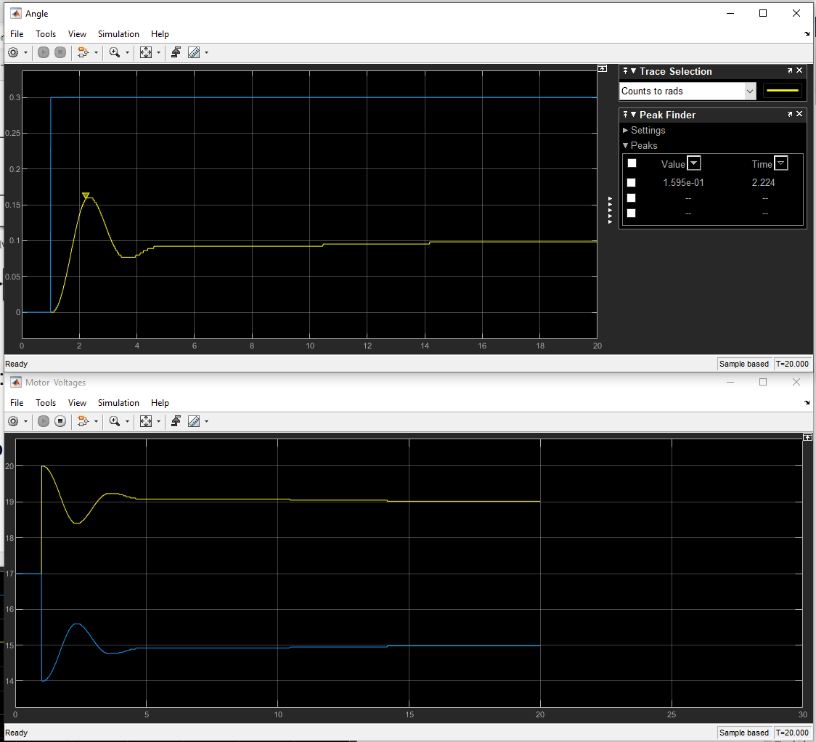
כמו כן, השתמשנו בפונקציית הPidTuner במטלאב כמצורף בקוד הנ"ל וכן מצאנו שאכן תוך כוונון ידני לא ניתן לספק את שתי הדרישות.

% Define the open-loop transfer function

G\_s = tf([1], [1, 5, 10]);

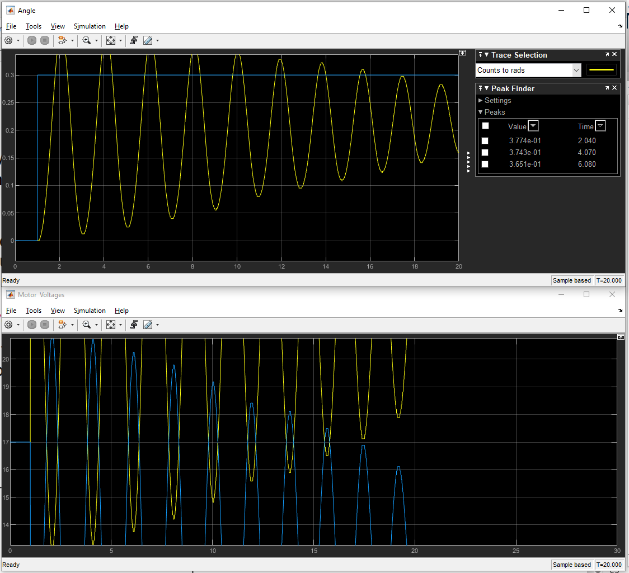
% Open the PID Tuner for interactive tuning

pidTuner(G\_s);

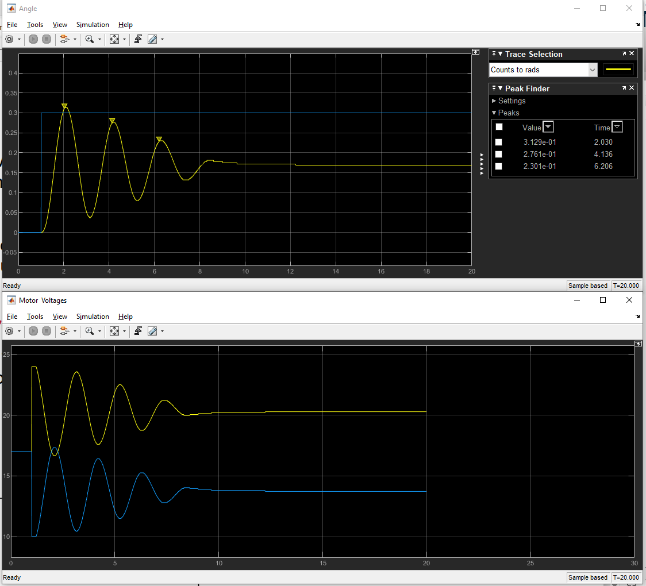
In-Lab Exercises

* 1. **Proportional Control**
     1. ביצענו Build והרצנו את הQuarc. קיבלנו את התגובה הבאה:
     2. הרצנו את הGain בין ערכים שונים של 10 עד 100 וקיבלנו את התוצאות הבאות:

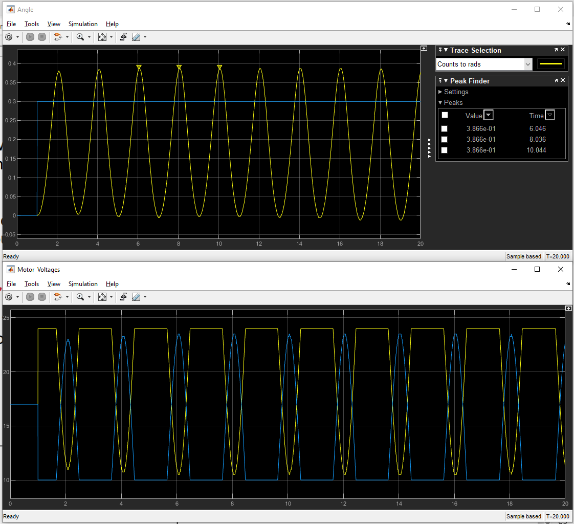
**K=10** – מופיע בסעיף הקודם



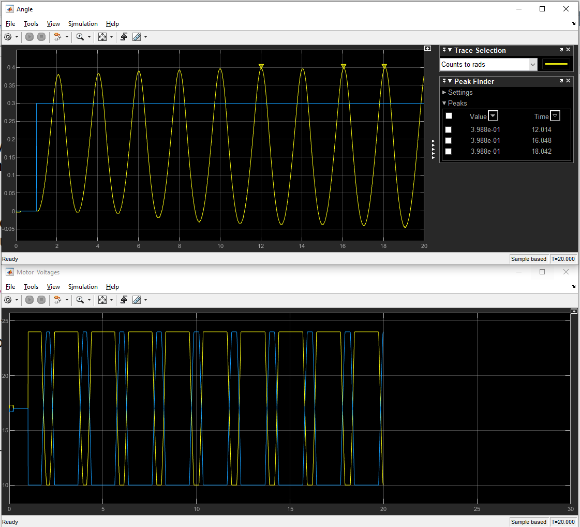
**K=50**



**K=25**



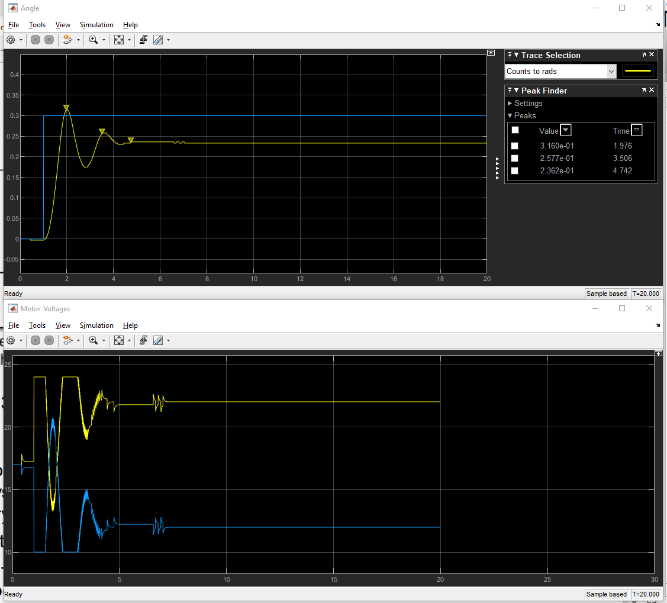
**K=100**



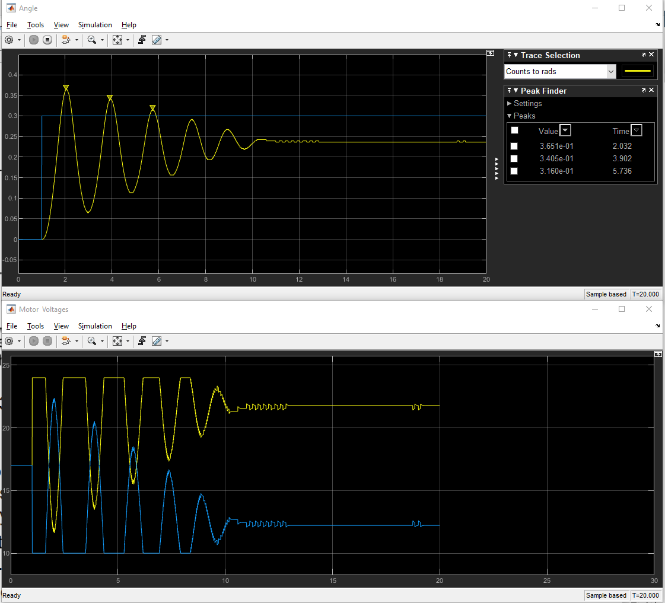
**K=75**

\*ניתן לראות לפי הגרפים שככל שהעלינו את הGain של הProportional קיבלנו יותר תנודות עד להגעת המצב היציב או אפילו – ללא הגעה למצב היציב (יציאה מיציבות המערכת) כפי שניתן לראות בK הגדולים יותר.

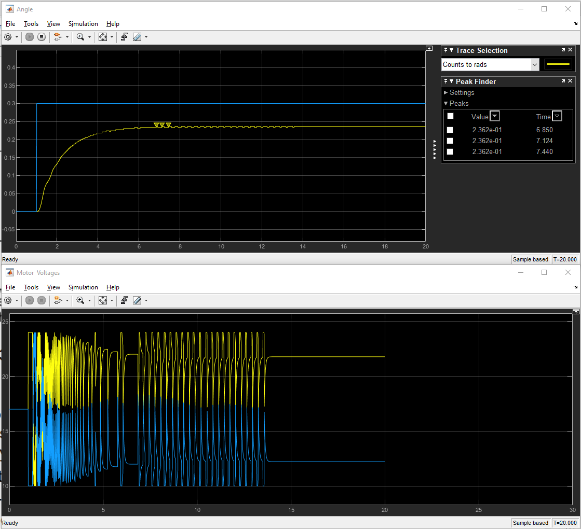
* + 1. כפי שניתן לראות בגרפים, כאשר עסקנו ב (הנמוך ביותר) זמן העלייה שלנו היה גם הגבוה ביותר (2.22 sec) וככל שהגדלנו את כך זמן העלייה קוצר (כצפוי). **אולם,** ככל שהגדלנו את ולפי המסקנה מהסעיף הקודם – המערכת נעשתה "פחות יציבה" עד כדי מצב של אי-יציבות כלל. על כן, לא נוכל לבצע זמן עלייה של עד 1.1 שניות במערכת בחוג הסגור ליציבות.
  1. **Derivative Control**
     1. הצבנו את הProportional Gain לערך של 75 קבוע, וביצענו מספר מדידות של ערך הderiative בין 0 ל100. להלן התוצאות:



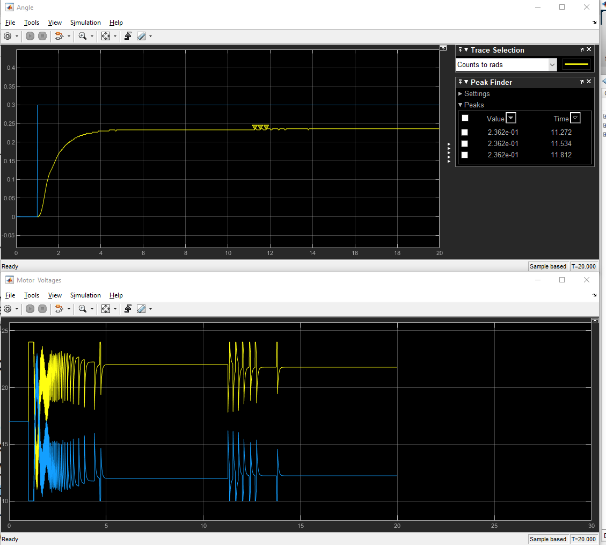
**K=10**



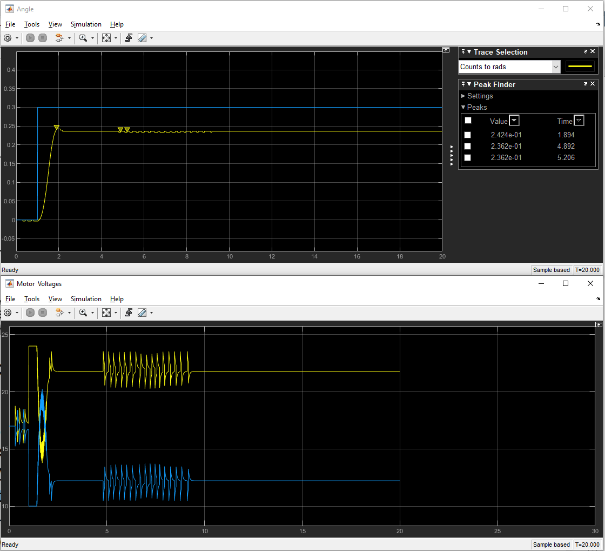
**K=2.5**



**K=100**

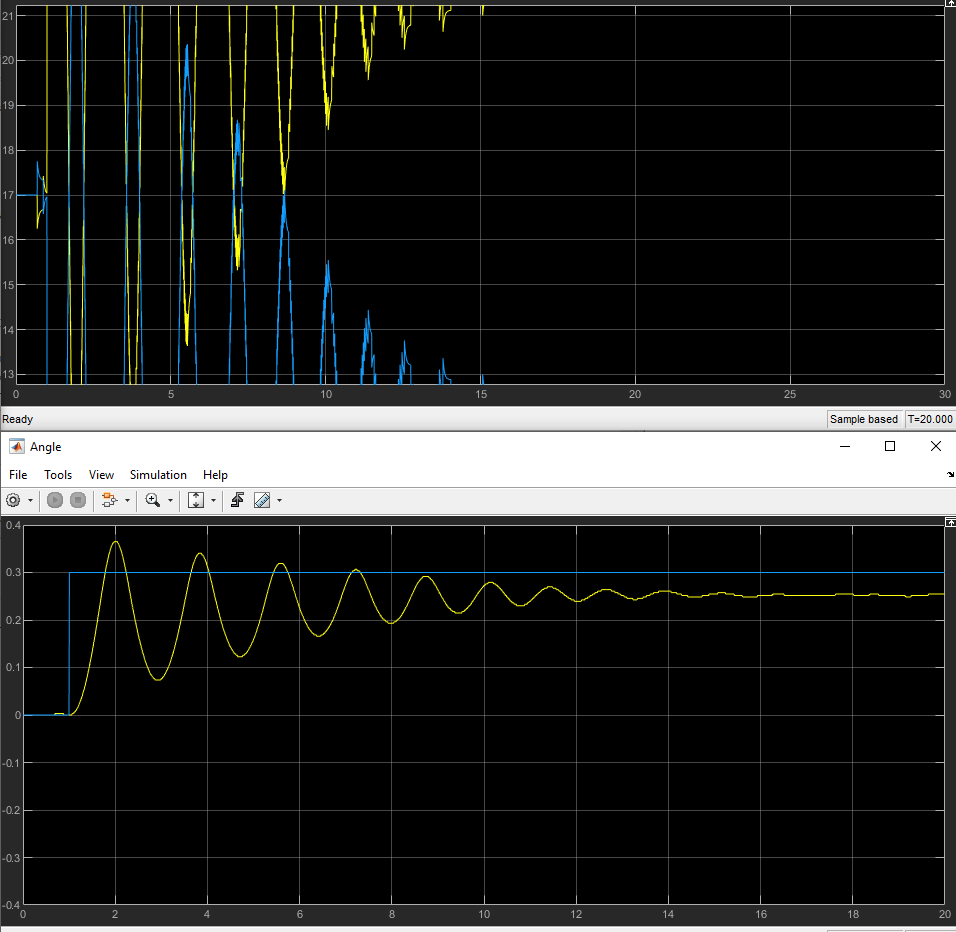


**K=65**

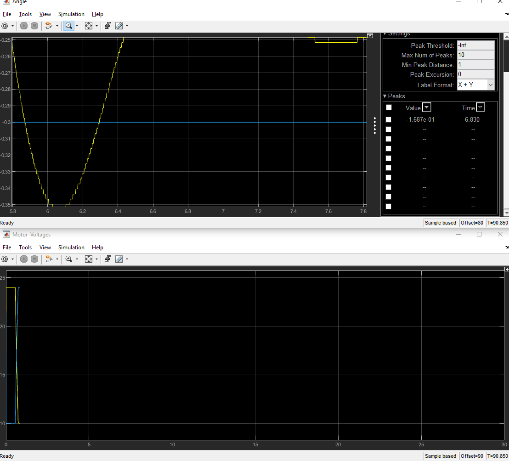


**K=25**

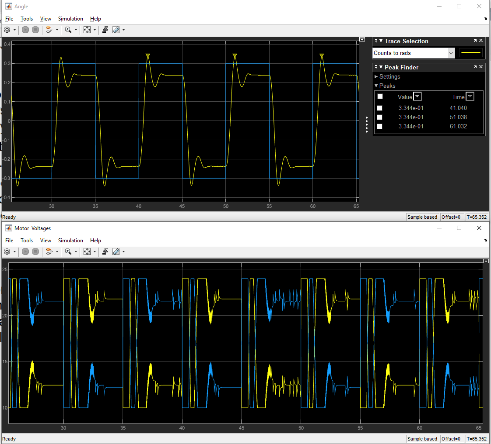
\*ניתן לראות שהגדלת גורם הderiative מצמצם את הOS שכן ככל שהגדלנו את הערך, קיבלנו Overshoot נמוך יותר בגרפים. כמו כן, ניתן לראות שזמן ההתייצבות התקצר גם כן.  
עבור קיבלנו את התוצאה הבאה:

 *בסעיף הקודם, קיבלנו שעבור ראינו שהמערכת אינה יציבה. כמו כן, לעיל ניתן לראות כי המערכת מגיעה לבסוף להתייצבות לאחר זמן ממושך. על כן, ניתן להסיק שקיים עבורו נגיע למצב של חוסר יציבות.*

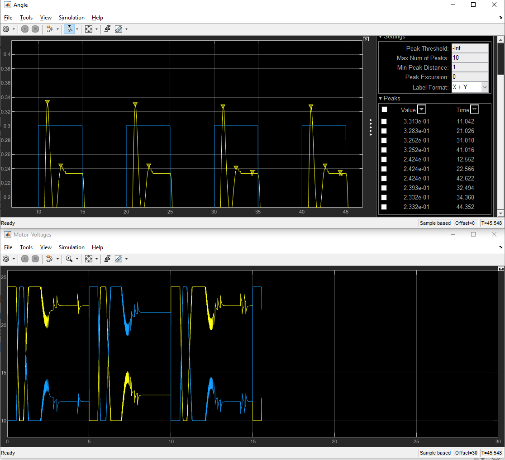
* + 1. ביצענו את השינוי לSquare-Wave ושינינו את זמן הריצה ל.
    2. הגענו לערך קריטי של עבורו הגל הריבועי לא חוצה את ה כפי שניתן לראות בגרפים הבאים:



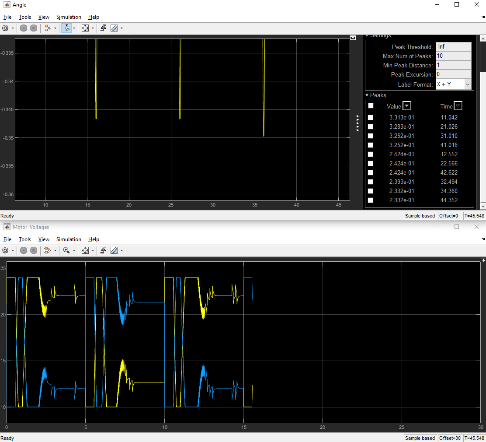
**K=16**ניתן לראות שחוצה ערך תחתון



**K=17**נראה כי לא חוצה בפלטים הבאים:

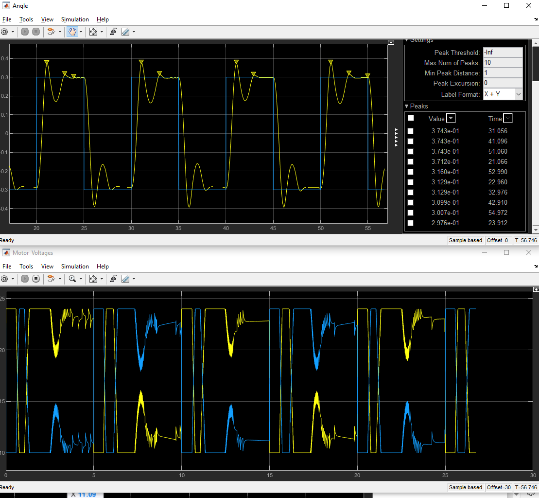


**K=17**ניתן לראות שלא חוצה ערך עליון

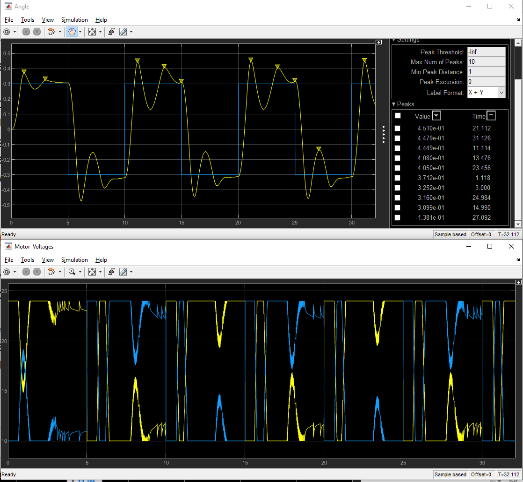


**K=17**ניתן לראות שלא חוצה ערך תחתון

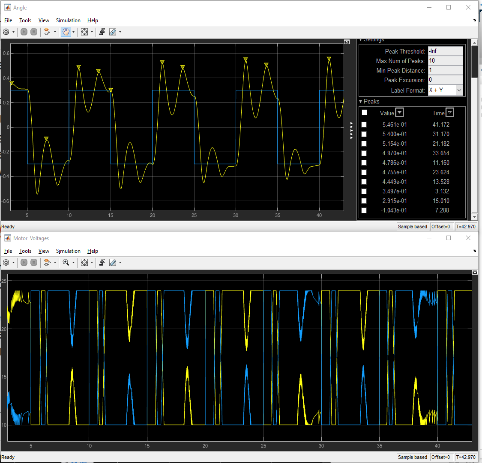
* 1. **Integral Control**
     1. השארנו את ערכי הK הקודמים, וביצענו מספר מדידות של ערך Integral בין 0 ל100. להלן התוצאות:



**K=25**



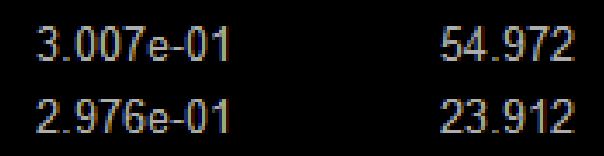
**K=50**

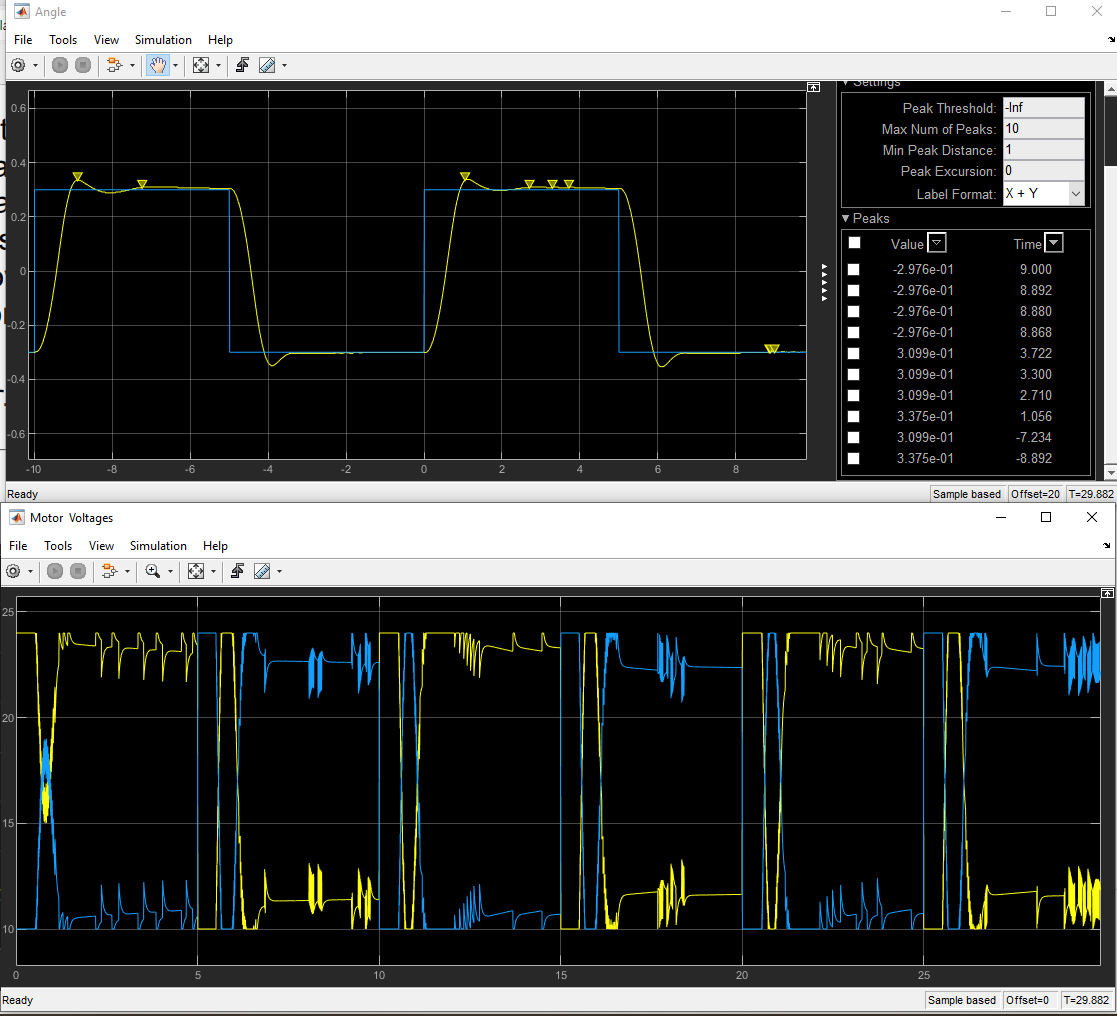


**K=100**

\*ניתן לראות שעבור הגדלת ערכי קיבלנו הגדלה של הOS וכן גדילה בערך הRising-Time, כמו כן, ככל שהגדלנו את ערך קיבלנו תגובה פחות יציבה ולכן גם נסיק שאם נגדיל את הערך הנ"ל יותר מדי – נצא מיציבות.

* + 1. ניתן לראות בסעיף קודם שעבור קיבלנו את הדרוש, שכן מצילום המסך של המדידות ניתן לראות שעומד בתנאים המצוינים:



* 1. **Response Tuning**

את הפלט הנ"ל קיבלנו מהערכים: ואכן ניתן לראות שעומדים בתנאים המצוינים:

**Aero Simulation**

Preparation questions

1. לפי דיאגרמת הבלוקים המוצגת לנו, ישנם 3 משתני מצב:
   1. Pitch – מייצג את זווית העלרוד והוא נכנס לPID Controller הימני לבקרת הזווית
   2. X Position – מייצג מיקום על ציר הX ונכנס לPID Controller הראשון
   3. Z Position
2. נראה את הכניסות והמוצאים לכל בקר PID:
   1. עבור Sway PID Controller:
      1. כניסות – Target X, X
      2. מוצאים – error, u\*  
         כאשר u מוזן לתוך בקרת (מסנן) Pitch limit וכן לתוך בקר הpitch כדי למצוא את הpitch הרצוי
   2. Throttle PID Controller:
      1. כניסות – Target Z, Z
      2. מוצאים – error, u\*  
         כאשר u מוזן לתוך בקרת (מסנן) ומשם לסכימה (ביחד עם הpitch) לטובת המתח הדרוש למנוע
   3. Pitch PID Controller
      1. כניסות – Pitch, u\*

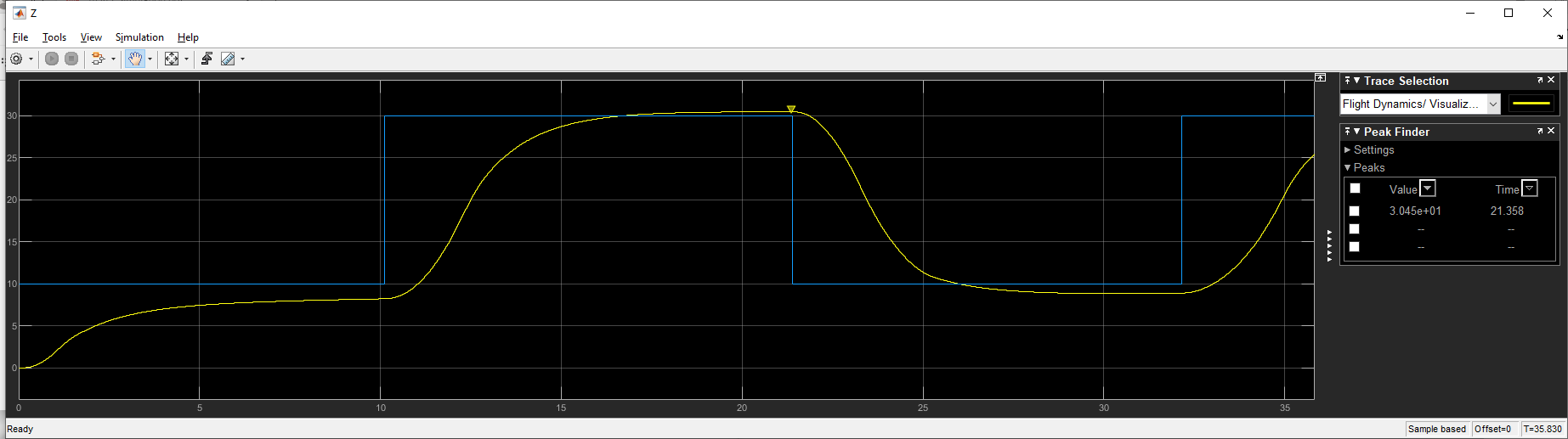
כאשר u\* הוא המוצא שתיארנו בSway

* + 1. מוצאים – error, u\*

כאשר u מוזן מועבר לסכימה (ביחד עם הu\* של הThrottle) לטובת המתח הדרוש למנוע

1. דיאגרמת בלוקים

In-Lab Exercises

* 1. **Throttle PID Controller**
     1. ביצענו את הDouble Clock על הPitch PID Controller ועדכנו את ערכי הPID כפי שנמצא בחלק 1
     2. וידאנו שאכן הסוויצ'ים מחוברים כהלכה
     3. ביצענו כוונון ידני לערכי הPID עבור הThrottle עד אשר הגענו לproper response אשר עומדים בתנאים: Settling Time < 10s, OS<25%. הפלט להלן:

ניתן לראות שאכן הOS קטן מ25% וכן שזמן ההתייצבות המתואר הינו קטן מ10 שניות.  
הערכים שהתקבלו:

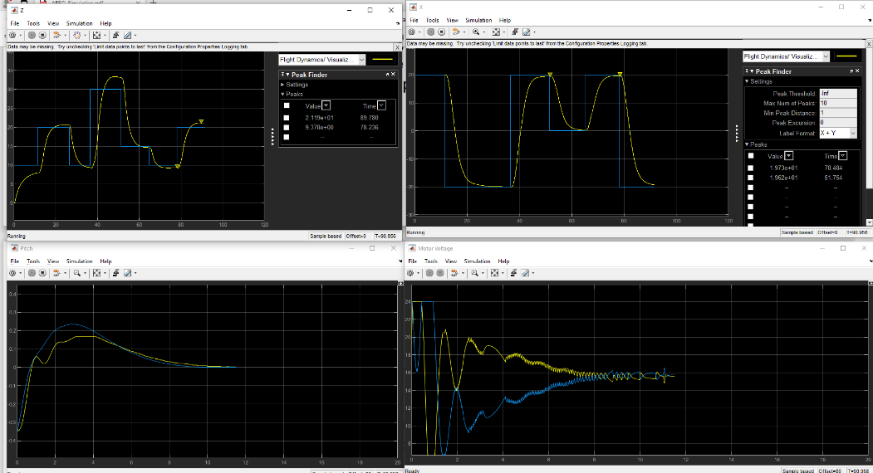
* + 1. ביצענו את המעבר סוויץ' לSway
    2. תמונה שמכילה תכונות מולטימדיה, תוכנה, תוכנה גרפית, צילום מסך

       התיאור נוצר באופן אוטומטיביצענו כוונון ידני לערכי הPID עבור הSway עד אשר הגענו לproper response אשר עומדים בתנאים: Settling Time < 10s, OS<25%. הפלט להלן:

ניתן לראות שאכן הOS קטן מ25% וכן שזמן ההתייצבות המתואר הינו קטן מ10 שניות.  
הערכים שהתקבלו: . אכן השתמשנו בPD ללא ה*.*

לגבי זמן התגובה בין הלולאה הפנימית לחיצונית, נוכל להסיק כי ההנחה עבור Cascade Controller Configuration הינה שהלופ הפנימי צריך להיות מהיר יותר מהלופ החיצוני, על מנת להשיג ביטול מהיר של ההפרעות לפני שהן משפיעות על הPLANT.

* + 1. ביצענו את הקונפיגורציה של הסווי'צים עבור הPath Tune Waypoints ובדקנו שילוב של הפלט עבור שני בקרי הPID (Throttle, Sway) במצב משולב. קיבלנו שעבור בדיקה שכזו ההנחה לParallel Control Configuration **אינה** נכונה שכן ייצוב כל בקר בפני עצמו, אינו מבטיח ייצוב של המערכת הכוללת כאשר עובדת באופן משולב.

על כן, מצאנו פרמטרים מתאימים עבור כל אחד מבקרי הPID לטובת מערכת שעומדת בקריטריונים, ממקמת את הרחפן כהלכה ובאופן יציב:

הערכים שהתקבלו עבור Throtlle: .

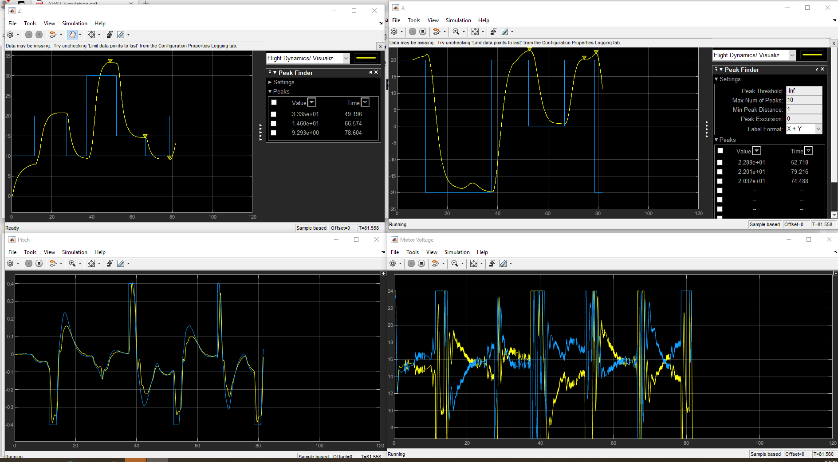
הערכים שהתקבלו עבור Sway: .

**בונוס:**

* + 1. הוספנו הפרעות "רוח" למערכת:



**Wind=10 m/s**



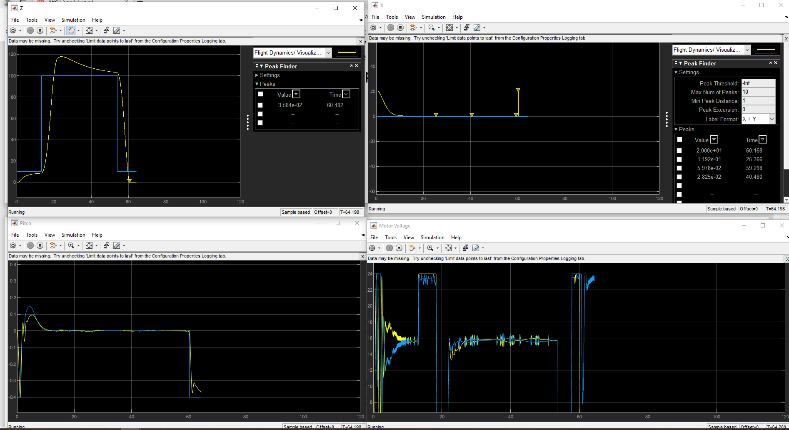
**Wind=25 m/s**

ניתן לראות שעבור שני הערכים השונים לWind קיבלנו שהמערכת עדיין שומרת על יציבותו של הרחפן עם **אותם** ערכים של בקרי הPID אליהם הגענו בסעיף הקודם.

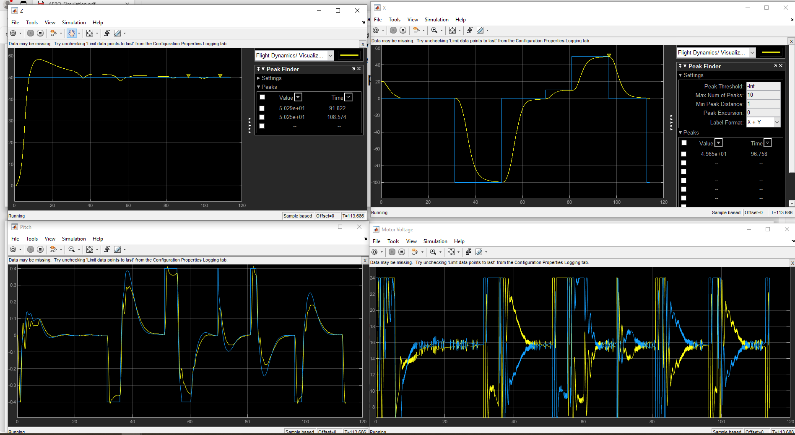
* + 1. שינינו את ערכי הPath-Waypoints ובדקנו את יציבות המערכת למס' קונפיגורציות שונות:



**[0 0 10; 50 0 25; -10 0 10; 20 0 30; 0 0 10];**



**[0 0 10; 0 0 100; 0 0 10; 0 0 100; 0 0 30];**



**[0 0 50; -100 0 50; 0 0 50; 10 0 50; 50 0 50];**

כאשר הצבנו את המערכת לקונפיגורציה הראשונה, קיבלנו שהיא מצליחה עדיין לשמור על יציבותה כפי שניתן לראות בפלט. בקונפיגורציה השנייה, הגדרנו "ערכי קיצון" בציר הZ וניתן לראות שהמערכת **לא** שמרה על יציבותה במקרה זה. בשונה מכך, בקונפיגורציה השלישית, הגדרנו "ערכי קיצון" בציר הX והמערכת **כן** שמרה על יציבותה במקרה זה.

**Appendix**

להשלים