חישוב הספק עבור רוכבי אופניים

מודל חישובי

מאת

נדב הלחמי

עבודה זו מוגשת כעבודה בהיקף של 1 יח' כמילוי חלקי של הדרישות לקראת קבלת ציון במדע חישובי

עבודה זו בוצעה בהדרכת <u>שלמה רוזנפלד</u>

יולי 2014

תמצית

בפרויקט "חישוב הספק עבור רוכבי אופניים" אנסה לענות על שאלה ובעיה העולה רבות בקרב רוכבי הכביש והשטח, האם ניתן לחשב את הספק הרוכב בלי להשתמש במד הספק? בפרויקט אחשב הספק עבור רוכב מסוים ואשווה אותו להספקו הנמדד באמצעות מד הספק.

תוכן עניינים

מבוא

הפרויקט "חישוב הספק עבור רוכבי אופניים" יעסוק בניסיון להגיע לנתוני הספק שנמדדו באמצעות מד הספק אמיתי באופן חישובי. ראשית כל אגדיר מספר מושגים שיעזרו בהבנת הפרויקט:

הספק- הנגזרת של העבודה W לפי הזמן t. ביחידת זמן נתונה ההספק הממוצע הוא כמות העבודה המושקעת או מתקבלת ביחידת זמן. העבודה אותה מבצע הרוכב היא הכוח שהוא מפעיל על מנת להניע את האופניים לאורך דרך מסוימת. לכן, ההספק הוא הכוח שהרוכב מפעיל על מנת להתקדם כפול המהירות שלו באותו רגע. מד הספק מסוגל למדוד את הכוח שמפעיל הרוכב על האופניים ואת המהירות שבה מפעיל הרוכב כוח זה. לדוגמה: אם ההספק נמדד על הפדלים אז ההספק יהיה הכוח שמפעיל הרוכב על הפדלים כפול מהירות הפדלים (היקף התנועה של הפדל חלקי הזמן לסיבוב אחד), הגדרה זו שקולה למכפלת מומנט הכוח במהירות הזוויתית. עבור מד הספק שכזה על הרוכב לשלם סכום לא מבוטל של למעלה מ-2000$. קיימות שתי דרכים למדידת הספק של רוכב אופניים: בדרך ישירה, מודדים את הכוח שהרוכב מפעיל על חלקים שונים באופניים ומכפילים אותו במהירות של התקדמות החלק שעליו מפעיל הרוכב את הכוח ומודדים אותו. בדרך עקיפה, מודדים את סך הכוחות המתנגדים להתקדמות הרוכב ואותם מכפילים במהירותו, כאשר דרך זו מתבססת על החוק השלישי של ניוטון. בדרך בה אחשב את הספק הרוכב אשתמש בגורמי התנגדות התלויים באופניים וברוכב ולכן אכפיל את סכומם במהירות הרוכב. להלן הסבר מתמטי לנוסחת ההספק:

את נוסחת ההספק ניתן להציג כך: $\frac{\Delta W}{\Delta t} = P$, מכיוון ש: $\Delta W = F\Delta x$ נגיע לנוסחה: $P = Fv = \frac{F\Delta x}{\Delta t}$

ΔW = העבודה אותה מבצע הרוכב.

Δt = יחידת זמן.

F = הכוח שמפעיל הרוכב בהנעת המערכת (רוכב+אופניים).

Δx = אורך דרך.

v = מהירות הרוכב.

P = הספק הרוכב.

ההתנגדות F מתחלקת לארבעה חלקים:

F1- הכוח שמתנגד לרוכב כשהוא מטפס ודוחף אותו כשהוא יורד, הוא הולך וגדל ככל שהשיפוע גדול יותר, ואותו אכנה "התנגדות הגרביטציה" (למרות השימוש במילה "התנגדות" הכוח הזה יכול להיות גם שלילי כאשר הרוכב בירידה, על כן הגרביטציה לא תמיד מתנגדת). את F1 אחשב לפי הנוסחה $F1 = m * g * \sin\alpha$ כאשר m זה מסת הרוכב+האופניים, g זה תאוצת הגרביטציה על כדור הארץ $(9.8\,{}^{m}/_{s^2})$ ו- $\sin\alpha$ זה שיפוע הדרך שאותו אחשב לפי נתוני הגובה הידועים לי.

F2- הכוח המתנגד לתנועה בתוך האוויר ואותו אכנה "התנגדות האוויר". את F2 אחשב לפי הנוסחה: $F_D = \frac{1}{2} \rho v^2 C_D A$ (מתוך הערך Drag(physics) בויקיפדיה), המייצגת את התנגדות האוויר עבור גוף מסוים כאשר ההתנגדות תלויה במהירות הרוכב ביחס לרוח ובצפיפות האוויר. צפיפות האוויר ρ תלויה בגובה ובטמפרטורה, וחישובה יתואר בפירוט בהמשך. המהירות **ביחס לקרקע** נתונה לי (נמצאה באמצעות מד המהירות של הרוכב). הקבוע C_D מייצג את מקדם הגרר של הגוף והקבוע A מייצג את שטח החתך של הגוף, אייצג אותם כקבוע אחד k1 התלוי ברוכב ומציאתו תפורט בהמשך. סיבה מרכזית לאי-דיוקים שיתקבלו בעת חישוב ההספק היא המחסור במידע על מהירות הרוכב **ביחס לאוויר**.
בחישוביי אתייחס למהירות הרוכב ביחס לקרקע כמהירותו ביחס לאוויר, מה שיביא ככל הנראה לתוצאות מדויקות ברכיבות ללא רוח אך ברכיבות עם רוח תתקבל שגיאה.
F3- ההתנגדות הקבועה, תלויה באופניים ולא תלויה במהירות. ההתנגדות הקבועה נובעת מפרמטרים רבים היוצרים התנגדות לרוכב, כמו חיכוך השרשרת בגלגלי השיניים, זרימת אוויר בתוך הצמיגים וכו'. מציאת ההתנגדות הקבועה, שאותה אייצג באמצעות הקבוע k2, תפורט בהמשך.
F4- סכום הכוחות F1+F2+F3 כפול המהירות שווה לעבודה שעל הרוכב להשקיע על מנת לרכוב במהירות הזאת. במידה והוא משקיע מעבר לכך הוא מאיץ. במידה והוא משקיע פחות מכך הוא מאט. מכפלת התאוצה (או התאוטה) במסה של הרוכב+האופניים ובמהירות תיתן לנו את הפער בין עבודת הרוכב לבין $(F1 + F2 + F3) \times V$. אסמן ב-F4 את הכוח הנדרש לתאוצה ואקרא לו "התנגדות התאוצה" (למרות השימוש במילה "התנגדות" הכוח הזה יכול להיות גם שלילי כאשר מדובר בתאוטה, על כן הכוח הזה לא תמיד מתנגד). את התנגדות התאוצה אחשב באמצעות הנוסחה $F4 = m * a$ (לפי החוק השני של ניוטון), כאשר m זה מסת הרוכב+האופניים ו-a זה תאוצת הרוכב שתחושב על פי הנוסחה: $a = \frac{\Delta v}{\Delta t}$.
סכום כל ההתנגדויות האלה כפול מהירות הרוכב יהיה **הספק הרוכב המחושב**, שאותו אשווה **להספק שנמדד** באמצעות מד הספק, השוואה זו תיתן לי הערכה איכותית להצלחת הפרויקט.

K1 ו-K2

מי אלה k1 ו-k2 ומה תפקידם?

על מנת להבין מי אלה k1 ו-k2 עלינו להבין ממה מורכב ההספק אותו אנו מחשבים. ההספק אותו אנו מחשבים מורכב מ4 כוחות שמכפלתם במהירות תיתן את ההספק, אלו הם ארבעת הכוחות:

$F1 = m * g * incline[i]$ הוא תלוי רק בשיפוע ובשני קבועים ידועים.

$F4 = m * \text{acceleration[i]}$ הוא תלוי רק בתאוצה ובקבוע אחד ידוע.

$F2 = \frac{1}{2}\rho v^2 * k1$ הוא תלוי בצפיפות האוויר ובמהירות אך גם בעוד קבוע ואנו ננסה להבין מדוע נוסף הקבוע k1 למשוואה. ניקח לדוגמה רוכב ענק מימדים (גובה 2.5 מטר, משקל 150 ק"ג) ולצידו רוכב קטן מימדים (1.5 מטר, משקל 50 ק"ג), למי מהם יהיה יותר קשה לרכוב דרך הרוח? מי מהם ייצר יותר התנגדות רוח? ללא ספק לרוכב הענק יהיה יותר "קשה" ופה בא לידי ביטוי k1 במשוואה, ככל שהרוכב יותר אירודינמי (בעל התנגדות נמוכה יותר לאוויר) כך ה-k1 שלו יהיה נמוך יותר.

$F3 = k2$ הוא ההתנגדות הקבועה שלא תלויה במהירות וננסה להבין מדוע יש צורך בה . ניקח לדוגמה אופניים שלחץ האוויר בגלגליהם הוא 40 psi ולצידם אופניים שלחץ האוויר בהם הוא 120 psi, אילו אופניים ייסעו יותר בקלות? לרוכב על איזה אופניים יהיה יותר "קל"? האופניים עם לחץ האוויר הגדול יותר יתקדמו יותר בקלות מכיוון שבלחץ אוויר נמוך נוצרות זרימות פנימיות של אוויר בגלגל ובנוסף אזורים בעלי צפיפות משתנה באוויר בגלגל, כתוצאה מכך יתפתח חום בגלגל שהוא בזבוז של אנרגיה. דוגמה נוספת היא החיכוך בין השרשרת לגלגלי השיניים, כתוצאה מחיכוך זה נוצרת אנרגית חום שהיא בזבוז של אנרגיה מצד הרוכב. גורמים אלה יחד עם עוד גורמים רבים אחרים התלויים באופניים הם גורמים שצריך לקחת בחשבון בעת חישוב הספק. לצורך העניין אני בטוח שכל רוכב שהתנסה ברכיבת כביש ושטח יודע שעל משטח כביש ישר לאופני הכביש יהיה הרבה יותר קל להתקדם מאשר לאופני השטח. על כן יש צורך בגורם התנגדות נוסף וקבוע, התלוי באופניים, והוא k2.

איך גיליתי מהם k1 ו-k2 עבור הרוכב שנתוני הרכיבה שייכים לו?

על מנת למצוא את k1 ו-k2 השתמשתי בשיטת הריבועים הפחותים: **שיטת הריבועים הפחותים** היא שיטת אומדן סטטיסטית, שבה משערכים גודל לא ידוע מתוך קבוצת תוצאות מדודות כלשהן. (מתוך ויקיפדיה) בעזרת שיטת הריבועים הפחותים אני מוצא את k1 ו-k2 עבורם סכום ריבועי השגיאות בין ערכי ההספק הנמדד לערכי ההספק המחושב ע"פ משוואת ההספק $P = (F1 + F2 + F3 + F4) * V$ הוא מינימאלי.

האלגוריתם

האלגוריתם בעיקרו מחולק לחמישה חלקים:

1. מציאת "התנגדות הגרביטציה" f1 באמצעות אומדן הגובה ומציאת השיפועים המשתנים. על מנת למצוא את f1 האלגוריתם עושה שימוש בשיטות: approximation, get_incline, get_f1.
2. מציאת התנגדות האוויר וההתנגדות הקבועה, f2 ו-f3. על מנת למצוא את f2 ו-f3 האלגוריתם עושה שימוש בשיטות get_ro וget_f2f3.
3. מציאת "התנגדות התאוצה" f4. על מנת למצוא את f4 האלגוריתם עושה שימוש בשיטות get_acceleration ו-get_f4.
4. באמצעות חלקי ההתנגדות השונים שנמצאו בשלבים 1-3 יצירת רשימת הספקים mywatts באמצעות השיטה get_mywatts. לרשימת ההספקים שהתקבלה מחשבים ממוצעים נעים באמצעות השיטה approximation. את הרשימה החדשה הופכים לחיובית תמיד (הספק שלילי מתקבל בעת לחיצה על ברקסים, מכיוון שמד ההספק לא מראה הספק שלילי אנו נאפס את כל התוצאות השליליות).
5. השוואה גרפית ומתמטית בין רשימת ההספקים הנמדדים (ממד הספק אמיתי) לבין רשימת ההספקים שחושבו avgmywatts. ההשוואה המתמטית נעשית ע"פ השיטה check שמבוססת על "מתאם פירסון", וההשוואה הגרפית באמצעות הפונקציה plot.

הסבר השיטות הבסיסיות

השיטה avg:

תמצית: השיטה מקבלת רשימה ומחזירה את ממוצע ערכיה.

```python
def avg (lst):
    sumlist=0
    for i in lst:
        sumlist=sumlist+i
    return float(sumlist)/len(lst)
```

$$\frac{sum(list)}{\mathrm{len(list)}} = \mathrm{avg(list)}$$

הסבר: השיטה מגדירה משתנה sumlist ומאתחלת אותו בערך 0. תפקידו יהיה לסכום את כל איברי הרשימה (נעשה באמצעות לולאת for). השיטה מחזירה את sumlist חלקי אורך הרשימה. המספר המוחזר הוא מטיפוס float והוא ממוצע ערכי הרשימה שהתקבלה כפרמטר.

השיטה approximation:

תמצית: השיטה מקבלת רשימה וגודל "חלון" ומחזירה את רשימת ממוצעים נעים של הרשימה. כל ערך ברשימה הוא ממוצע של חצי החלון איברים לפניו וחצי החלון איברים אחריו. הרשימה החדשה היא בגודל הרשימה המקורית כאשר גודל החלון חלקי 2 האיברים הראשונים והאחרונים הם המקוריים מהרשימה המקורית.

```python
def approximation (lst,w): #list,window(int)
    a=[avg(lst[i-w/2:i+w/2]) for i in range(w/2,len(lst)-w/2)]
    newlist=[lst[i] for i in range(w/2)]
    newlist=newlist+a
    newlist=newlist+[lst[i] for i in range(len(lst)-w/2,len(lst))]
    return newlist
```

הסבר: השיטה מגדירה רשימה a ששומרת את איברי הרשימה הממוצעת מ-w/2 ועד אורך הרשימה שהתקבלה כפרמטר lst פחות w/2. הגדרת כל איבר ברשימה a נעשית באמצעות ממוצע (השיטה avg) של מספר ערכים שהתקבל כפרמטר w. לדוגמה האיבר ה-0 ברשימה a יהיה ממוצע w האיברים הראשונים ב-lst, האיבר ה-1 ברשימה a יהיה ממוצע האיבר ה-1 ב-lst עד האיבר ה-w+1 ב-lst וכך הלאה. כתוצאה מכך הרשימה a היא רשימה באורך len(lst)-w. מכיוון שאנו רוצים להחזיר רשימה באורך הרשימה שהתקבלה, w/2 האיברים הראשונים ב-a ו-w/2 האיברים האחרונים ב-a יהיו האיברים המקוריים מ-lst.

השיטה get_incline:

תמצית: השיטה מקבלת רשימת גבהים ורשימת מרחקים (רשימה עולה לפי הדרך שרכב הרוכב מנקודת ההתחלה) ומחזירה את השיפוע בכל נקודה.

```
def get_incline(altlist,distlist):
    incline=[0]
    for i in range(1,len(altlist)):
        dalt=altlist[i]-altlist[i-1]
        ddist=distlist[i]-distlist[i-1]
        if ddist==0:
            incline.append(0)
        else:
            incline.append(dalt/ddist)
    return incline
```

הסבר: השיטה מגדירה רשימת שיפועים incline ונותנת לה ערך התחלתי [0]. באמצעות לולאת for הפונקציה מחשבת את השיפוע בכל נקודה ברכיבה באמצעות הפרש הגבהים חלקי הפרש המרחקים. על מנת למנוע חלוקה ב-0 אם הפרש המרחקים הוא 0 השיטה מוסיפה לרשימה incline את הערך 0. השיטה מחזירה את incline.

השיטה get_acceleration:

תמצית: השיטה מקבלת רשימת מהירויות ומחזירה רשימת תאוצות בכל רגע.

```
def get_acceleration(spdlist):
    acceleration=[(spdlist[i+1]-spdlist[i-1])/2 for i in range(1,len(spdlist)-1)]
    for i in range(2):
        acceleration.append(0)
    return acceleration
```

הסבר: השיטה מגדירה רשימת תאוצות acceleration. חישוב התאוצה נעשה באמצעות הפרש מהירויות חלקי הפרש זמנים. הפרש המהירויות מחושב ע"פ רשימת המהירויות spdlist שהתקבלה כפרמטר. הפרשי הזמן בין דגימות המהירות הם קבועים ואורכם שנייה אחת ועל כן הפרשי המהירויות תמיד מתחלק ב-2. acceleration מציגה רשימת תאוצות ביחידות $^{m}/_{s^2}$. על מנת שהרשימה המוחזרת תהיה שווה באורכה ל-spdlist השיטה מוסיפה עוד שני ערכי 0 בסוף הרשימה acceleration. השיטה מחזירה את הרשימה acceleration.

השיטה get_pressure:

תמצית: השיטה מקבלת רשימת גבהים ומחזירה רשימת לחצי אוויר עבור כל גובה.

```
def get_pressure(altlist):
    p0=101325
    l=0.0065
    t0=288.15
    g=9.80665
    m=0.0289644
    r=8.31447
    p=[p0*(1-l*h/t0)**((g*m)/(r*l)) for h in altlist]
    return p
```

הסבר: השיטה יוצרת רשימת לחצי אוויר p שנקבעת ע"פ רשימת הגבהים שמתקבלת כפרמטר. לחץ האוויר ישמש את האלגוריתם אחר כך במציאת צפיפות וf2 (התנגדות האוויר). חישוב p יעשה באמצעות הנוסחה הברומטרית: $p = p_0 \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R \cdot L}}$ המורכבת ממספר קבועים ופרמטר אחד (הגובה). נתוני הקבועים הוגדרו ע"פ הטבלה הבאה (מתוך הערך Atmospheric pressure בויקיפדיה):

Parameter	Description	Value
p_0	sea level standard atmospheric pressure	101325 Pa
L	temperature lapse rate, = g/c_p for dry air	0.0065 K/m
c_p	constant pressure specific heat	~ 1007 J/(kg•K)
T_0	sea level standard temperature	288.15 K
g	Earth-surface gravitational acceleration	9.80665 m/s^2
M	molar mass of dry air	0.0289644 kg/mol
R	universal gas constant	8.31447 J/(mol•K)

השיטה מחזירה את רשימת לחצי האוויר p.

הסבר השיטות המורכבות

השיטה get_f1:

תמצית: השיטה מקבלת את מסת הרוכב, תאוצת הגרביטציה (9.8), רשימת שיפועים ורשימת מספרים לפי מספר הדגימות. השיטה מחזירה את רשימת התנגדות הגרביטציה f1.

```
def get_f1(m,g,inclinelist,numslist):
    f1=[m*g*inclinelist[i] for i in numslist]
    return f1
```

הסבר: השיטה מחשבת את רשימת התנגדות הגרביטציה f1 ע"פ החוק השני של ניוטון באמצעות הכפלת השיפוע בכל נקודה במסת הרוכב+האופניים m ובתאוצת הגרביטציה g. רשימת השיפועים חושבה קודם לכן באמצעות השיטה get_incline. השיטה מחזירה את f1.

השיטה get_f4:

תמצית: השיטה מקבלת רשימת תאוצות ומסת הרוכב ומחזירה את התנגדות התאוצה (לפי החוק השני של ניוטון).

```
def get_f4(accelerationlist,m):
    f4=[accelerationlist[i]*m for i in range(len(accelerationlist))]
    return f4
```

הסבר: השיטה מחשבת את רשימת התנגדות התאוצה f4 ע"פ החוק השני של ניוטון באמצעות הכפלת התאוצה בכל נקודה acceleration במסת הרוכב+האופניים m. רשימת התאוצות חושבה קודם לכן באמצעות השיטה get_acceleration. השיטה מחזירה את f4.

השיטה get_ro:

תמצית: השיטה מקבלת טמפרטורה (במעלות קלווין) ורשימת גבהים ומחזירה רשימת צפיפויות אוויר.

```
def get_ro(t,altlist): #kelvin
    rSpecific=287.058
    ro=[i/(rSpecific*t) for i in get_pressure(altlist)]
    return ro
```

הסבר: השיטה יוצרת רשימת צפיפות אוויר ro שנקבעת ע"פ רשימת הגבהים שמתקבלת כפרמטר altlistו וע"פ הטמפרטורה באותו רגע t המתקבלת כפרמטר (הטמפרטורה במעלות קלווין ויצאתי מנקודת הנחה שהטמפרטורה לא משתנה במהלך הרכיבה). צפיפות האוויר תשמש את האלגוריתם אחר כך f2 (התנגדות האוויר). חישוב ro יעשה באמצעות הנוסחה: $$\rho = \frac{p}{R_{\text{specific}} T}$$ המורכבת מקבוע אחד rSpecific ושני פרמטרים p וtו. הקבוע rSpesific נלקח מתוך הערך density of air בויקיפדיה. T זו הטמפרטורה באותה עת (במעלות קלווין) ו-p זו רשימת לחץ האוויר שחושבה קודם לכן באמצעות השיטה get_pressure. השיטה מחזירה את רשימת צפיפות האוויר ro.

השיטה get_f2f3:

תמצית: השיטה מקבלת רשימת צפיפויות אוויר, רשימת מהירויות, רשימת מספרים לפי מספר הדגימות קבוע התנגדות האוויר (k1) וקבוע ההתנגדות הנוסף (k2) ומחזירה את התנגדות האוויר וההתנגדות הקבועה עבור כל דגימה.

```
def get_f2f3(rolist,spdlist,numslist,k1,k2):
    f2=[0.5*rolist[i]*spdlist[i]**2*k1 for i in numslist]
    f3=k2
    return f2,f3
```

הסבר: חישוב f2 נעשה באמצעות הנוסחה (מתוך הערך Drag(physics) בויקיפדיה):

$$F_D = \tfrac{1}{2}\rho v^2 C_D A$$ הנוסחה מורכבת משני פרמטרים ומשני קבועים (המייצגים ביחד קבוע אחד-k1). A הוא שטח החתך של הרוכב+האופניים ו-C_D זה מקדם הגרר של הרוכב+האופניים, את שניהם ביחד הצגתי באמצעות הקבוע k1. ρ הוא צפיפות האוויר באותו רגע שנלקחת מרשימת צפיפות האוויר rolist שחושבה קודם לכן באמצעות השיטה get_ro. v הוא מהירות הרוכב ביחס לאוויר (מכיוון שאין ברשותי מד רוח נאלצתי להזניח את מהירות הרוח ולהתייחס למהירות הרוכב ביחס לאוויר כמהירות הרוכב ביחס לקרקע) והוא נלקח מהרשימה spdlist. f3 הוא מספר קבוע שערכו הוא k2 (קבוע ההתנגדות הנוסף). על k1 ו-k2 אפרט בהמשך אך כרגע נתייחס אליהם כאל שני קבועים נתונים.

השיטה get_mywatts:

תמצית: השיטה מקבלת את כל ההתנגדויות, סוכמת אותן ומכפילה במהירויות עבור כל דגימת מהירות ומחזירה רשימת הספקים מחושבים.

```
def get_mywatts(f1list,f2list,f3,f4list,spdlist,numslist):
    mywatts=[(f1list[i]+f2list[i]+f3+f4list[i])*spdlist[i] for i in numslist]
    return mywatts
```

הסבר: חישוב ההספק נעשה ע"פ הנוסחה $P(t) = F \cdot v.$ מתוך הערך power(physics) בויקיפדיה. f הוא סכום כל ההתנגדויות (f1-4) וv זה מהירות הרוכב. דגימת הספק מורכבת מדגימה מכל אחד ממרכיבי ההתנגדות כפול דגימה מתאימה מרשימת המהירויות. לדוגמה, הערך mywatts[1000] הוא ערך ההספק המחושב בשנייה ה1000 (מכיוון שדגימות המהירות הן בהפרשים של שנייה ודגימות ההתנגדויות תואמות לכך). הוא מחושב באמצעות:
(f1list[1000]+f2list[1000]+f3+f4list[1000])*spdlist[1000]. השיטה מחזירה את רשימת ההספקים המחושבים mywatts.

הסבר שיטות נוספות

השיטה check:

תמצית: השיטה מקבלת שתי רשימות ומחזירה ערך בין 1 למינוס 1 על פי מתאם פירסון.

```
def check (xlist,ylist):
    sumlist=0
    averagex=avg(xlist)
    averagey=avg(ylist)
    mone=0
    mehanex=0
    mehaney=0
    for i in range(len(xlist)):
        mone=mone+(xlist[i]-averagex)*(ylist[i]-averagey)
    for i in range(len(xlist)):
        mehanex=mehanex+(xlist[i]-averagex)**2
        mehaney=mehaney+(ylist[i]-averagey)**2
    mehane=sqrt(mehanex*mehaney)
    return float(mone)/mehane
```

הסבר: מתוך הערך מתאם פירסון (מקדם הקורלציה) בויקיפדיה:

מתאם פירסון, או בשמו המלא **מקדם המתאם של פירסון**, הוא מדד למתאם לינארי בין שתי קבוצות של מספרים. כאשר מדובר בעיבוד נתונים סטטיסטי, ההתייחסות בדרך כלל היא לקשר סימטרי בין שני משתנים. ערכי המדד נעים בין (‎-1) לבין (‎+1) והם מסומנים באות R או ב-ρ:

- במתאם של ‎+1 מתקיים קשר חיובי מלא בין שני המשתנים.
- במתאם של ‎-1 מתקיים קשר שלילי מלא בין שני המשתנים.
- מתאם של 0 פירושו שבין שני המשתנים אין שום קשר לינארי.

במקרים רבים ימצאו קשרים בערכי ביניים, לדוגמה: מתאם של ‎+0.8 פירושו שקיים קשר חיובי בעוצמה חזקה.

מקדם המתאם של פירסון מספק מידע בשני מישורים:

1. עצמת הקשר בין המשתנים: ככל שהערך קרוב יותר ל ‎+1 או ל ‎-1 הוא חזק יותר.
2. כיוון הקשר בין המשתנים: ערך חיובי פירושו קשר חיובי. ערך שלילי פירושו קשר שלילי (הפוך).

השיטה check מחזירה ערך בין 1 למינוס 1 כך שאם הוא חיובי אז ככל שהוא קרוב יותר לאחד כך הקשר בין הרשימות xlist וylist שהתקבלו כפרמטרים חזק יותר. אם הוא שלילי אז ככל שהוא קרוב יותר למינוס אחד הקשר ההפוך בין הרשימות xlist וylist שהתקבלו חזק יותר. לדוגמה הקשר בין הרשימות [1,2,3] ו[1,2,3] יהיה 1 והקשר בין [1,2,3] ו[2,3,4] יהיה קרוב מאוד ל1. השיטה עצמה היא מימוש הנוסחה הבאה מתוך הערך בויקיפדיה:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

כך ש$\bar{x}$ הוא ממוצע ערכי x ו$\bar{y}$ הוא ממוצע ערכי y.

השימוש בשיטה check יהיה על מנת לבדוק כמה ערכי ההספק המחושבים קרובים להספק הנמדד.

השיטה doall:

תמצית: השיטה מקבלת את כל הנתונים הדרושים לחישוב הספק ומציירת גרף הספק מחושב לצד גרף הספק נמדד (עבור רכיבה ללא נתוני הספק ניתן לצייר רק את ההספק המחושב ובכך להשיג את מטרת הפרויקט). נוסף על הגרף השיטה מדפיסה את מתאם פירסון של רשימת ההספקים הנמדדים והמחושבים.

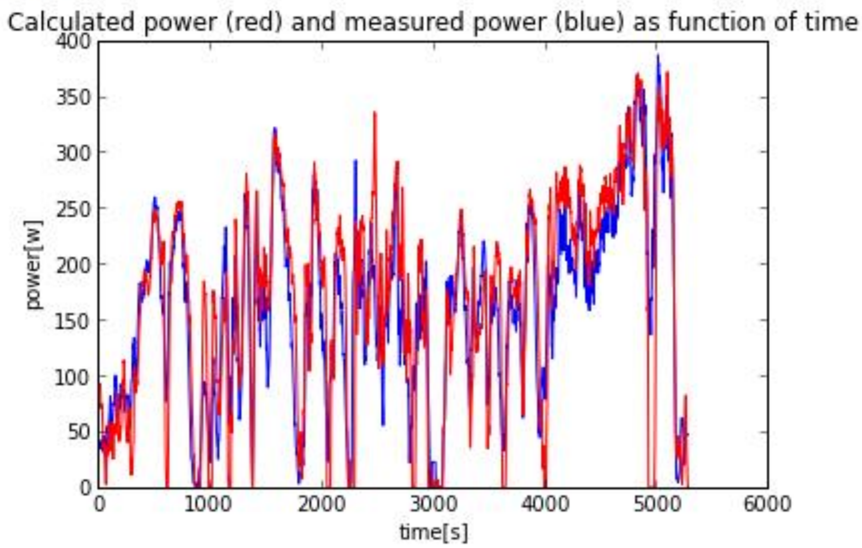
```
def do_all(numslist,spdlist,altlist,distlist,k1,k2,limits,m,wattslist):
    avgwatts=approximation(wattslist,limits)
    altitude=approximation(altlist,limits)
    incline=get_incline(altitude,distlist)
    f1=get_f1(m,9.8,incline,numslist)
    ro=get_ro(295.15,altitude)
    f2,f3=get_f2f3(ro,spdlist,numslist,k1,k2)
    acceleration=get_acceleration(spdlist)
    f4=get_f4(acceleration,m)
    mywatts=get_mywatts(f1,f2,f3,f4,spdlist,numslist)
    avgmywatts=approximation(mywatts,limits)
    k=0
    for i in avgmywatts:
        if i<0:
            avgmywatts[k]=0
        k+=1
    print check(avgwatts[limits/2:len(numslist)-limits/2],avgmywatts[limits/2:len(numslist)-limits/2])
    plot(numslist[limits/2:len(numslist)-limits/2],avgwatts[limits/2:len(numslist)-limits/2],'b-')
    plot(numslist[limits/2:len(numslist)-limits/2],avgmywatts[limits/2:len(numslist)-limits/2],'r-')
```

הסבר: תכליתה של השיטה doall היא להיות ידידותית למשתמש, היא משתמשת בכל השיטות שהזכרתי לעיל על מנת ליצור רשימת הספקים עבור המשתמש. הפעולה החישובית היחידה שהשיטה doall מבצעת היא לאפס את כל הערכים השליליים שהתקבלו בחישוב ההספק מכיוון שמד הספק אמיתי לא מראה ערכים שליליים. הסיבה ל"טעות בחישוב" היא לרוב לחיצה על הברקסים מצד הרוכב, מה שיוצר ערכי הספק שליליים.

הרצות שונות של השיטה doall:

```
do_all(nums,spd,alt,dist,0.136572389813,9.87323469208,35,93,watts)
```

```
0.862833305896
```

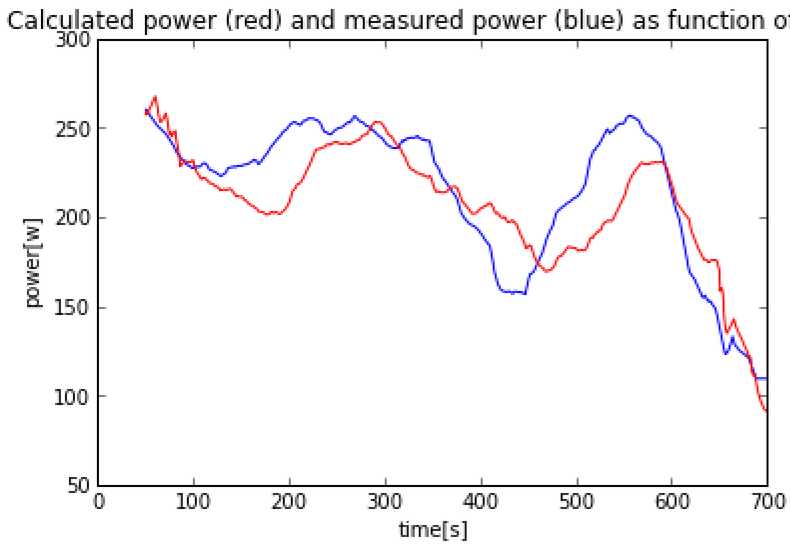


הסבר ההרצה: השיטה doall קיבלה (משמאל לימין): רשימת מספרים מ0 עד 5304 (לפי מספר הדגימות- במקרה זה 5305 דגימות), רשימת מהירויות (במטר לשנייה), רשימת גבהים (במטרים), רשימת מרחקים (כמה מטרים עבר הרוכב מתחילת הרכיבה), k1 של הרוכב+האופניים שהנתונים שהתקבלו שייכים לו, k2 של הרוכב+האופניים שהנתונים שהתקבלו שייכים לו, 35 (הערך שעל פיו יעשה אומדן הגבהים וההספקים, אותו w בשיטה approximation, 93 ק"ג (מסת הרוכב+האופניים ביחידות ק"ג), רשימת הספקים נמדדים.

השיטה הציגה שני גרפים על אותה מערכת צירים כאשר הגרף הכחול זה גרף ההספק הנמדד והגרף האדום זה גרף ההספק המחושב. הערך 0.86 הוא מתאם פירסון של ההספק המחושב והנמדד.

```
do_all(new_nums,new_spd,new_alt,new_dist,k1,k2,100,93,new_watts)
```

```
0.841549073708
```



***הרצה זו היא הרצה על ערכים חדשים לחלוטין מרכיבה נוספת של אותו רוכב. לא נעשה שימוש בנתונים אלה על מנת למצוא את k1 ו-k2.**

הסבר ההרצה: השיטה doall קיבלה (משמאל לימין): רשימת מספרים מ0 עד 749 (לפי מספר הדגימות- במקרה זה 750 דגימות), רשימת מהירויות (במטר לשנייה), רשימת גבהים (במטרים), רשימת מרחקים (כמה מטרים עבר הרוכב מתחילת הרכיבה), k1 של הרוכב+האופניים שהנתונים שהתקבלו שייכים לו (0.136 במקרה זה), k2 של הרוכב+האופניים שהנתונים שהתקבלו שייכים לו (9.873 במקרה זה), 100 (הערך שעל פיו יעשה אומדן הגבהים וההספקים, אותו w בשיטה approximation), 93 ק"ג (מסת הרוכב+האופניים ביחידות ק"ג), רשימת הספקים נמדדים.

השיטה הציגה שני גרפים על אותה מערכת צירים כאשר הגרף הכחול זה גרף ההספק הנמדד והגרף האדום זה גרף ההספק המחושב. הערך 0.84 הוא מתאם פירסון של ההספק המחושב והנמדד.

השיטה get_k1k2:

תמצית: השיטה מקבלת: גבול תחתון, גבול עליון, רשימת מספרים מ0 עד מספר הדגימות (דגימות הספק, מהירות), רשימת הספקים (בוואטים), רשימת מהירויות (מ'/ש'), רשימת התנגדויות גרביטציה והאצה ורשימת צפיפויות אויר. השיטה מחזירה את קבוע התנגדות האוויר (k1) וקבוע ההתנגדות הנוסף (k2).

```python
def get_k1k2(lowerLimit,upperLimit,numslist,wattslist,spdlist,f1list,f4list,rolist):
    a=[0.5*rolist[i]*spdlist[i]**2 for i in numslist[lowerLimit:upperLimit]]
    b=[1 for i in numslist[lowerLimit:upperLimit]]
    c=[]
    for i in numslist[lowerLimit:upperLimit]:
        if spdlist[i]==0:
            c.append(0)
        else:
            c.append(wattslist[i]/spdlist[i]-(f1list[i]+f4list[i]))
    #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    sa=0
    sb=0
    sc=0
    sd=0
    for i in numslist[:upperLimit-lowerLimit]:
        sa=sa+a[i]**2
        sb=sb+a[i]*b[i]
        sc=sb
        sd=sd+b[i]**2
    #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    num1=sd/(sa*sd-sb*sc)
    num2=-sb/(sa*sd-sb*sc)
    num3=-sc/(sa*sd-sb*sc)
    num4=sa/(sa*sd-sb*sc)
    #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    v1=[num1*a[i]+num2*b[i] for i in numslist[:upperLimit-lowerLimit]]
    v2=[num3*a[i]+num4*b[i] for i in numslist[:upperLimit-lowerLimit]]
    #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    k1=0
    k2=0
    for i in numslist[:upperLimit-lowerLimit]:
        k1=k1+v1[i]*c[i]
        k2=k2+v2[i]*c[i]
    #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    return k1,k2
```

הסבר: השיטה מוצאת את k1 ו-k2 על פי שיטת הריבועים הפחותים. השיטה הזו היא שיטה חד פעמית, תפקידה הוא למצוא את k1 ו-k2 עבור המשתמש. לאחר שהם נמצאו הוא יכול להשתמש באותם ערכים (בתנאי שלא החליף אופניים או שינה משהו בגופו בצורה משמעותית) על מנת לחשב את ההספק שלו ברכיבות נוספות, וזה מה שעשיתי בפרויקט שלי.

השתמשתי במשוואה זו $\hat{\boldsymbol\beta} = (X^TX)^{-1}X^T\boldsymbol y$ מתוך ויקיפדיה המחזירה את ערכי k1 ו-k2. המטריצה X במשוואה היא ערכי המקדמים של k1 ו-k2 והם ידועים. Y זה ערכי ההספק הנמדד פחות ההספק המחושב הידוע. הכפלת מקדמי k1 ו-k2 ב-k1 ו-k2 תיתן את ערכי Y הידועים וכך ניתן למצוא את k1 ו-k2 שהם $\hat{\boldsymbol\beta}$. את המשוואה ניתן להציג כך: $\begin{bmatrix} x1_1 & x2_1 \\ x1_2 & x2_2 \\ x1_n & x2_n \end{bmatrix} * \begin{bmatrix} k1 \\ k2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_n \end{bmatrix}$

דוגמה להרצת השיטה:

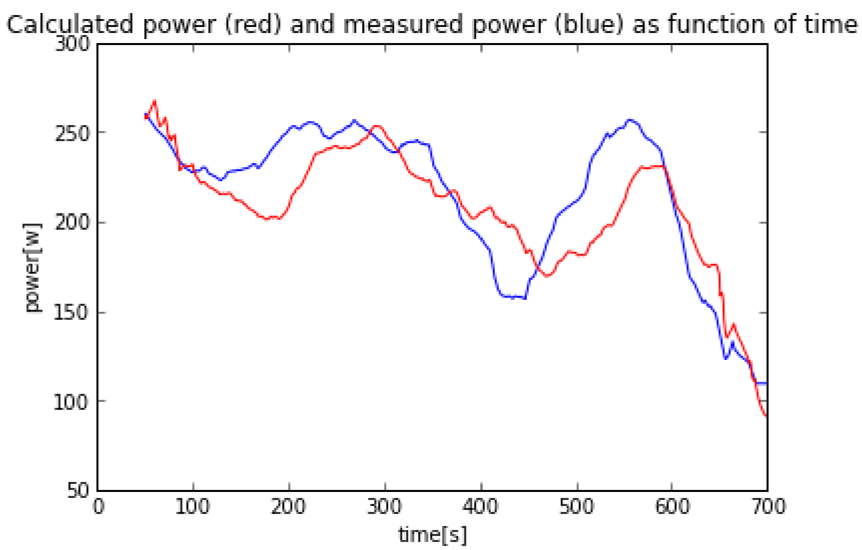
```
k1,k2=get_k1k2(limits/2,len(nums)-limits/2,nums,avgwatts,spd,f1,f4,ro)
print k1,k2
```

```
0.136572389813 9.87323469208
```

```
do_all(nums,spd,alt,dist,0.136572389813,9.87323469208,35,93,watts)
```

```
do_all(new_nums,new_spd,new_alt,new_dist,k1,k2,100,93,new_watts)
```

```
0.841549073708
```



הדוגמה השנייה היא מיוחדת מכיוון שבה חישוב ההספק נעשה **בלי שימוש כלל במד הספק**. הקבועים k1 ו-k2 חושבו מקבוצת מדידות אחת, בעזרתם חישבתי את הספק על קבוצת נתונים חדשה, אותו השוואתי להספק הנמדד. הגרף הכחול הוא גרף ההספק שנמדד והגרף האדום הוא גרף ההספק שחושב, מתאם פירסון של שתי רשימות אלו הוא 0.84. הגרפים האלו מייצגים רכיבה נוספת של אותו רוכב שלו חישבתי את k1 ו-k2 על פי רכיבה **אחרת** שלו, והמתאם הזה מוכיח את הצלחת הפרויקט שלי בדיוק גבוה יחסית. פה הושגה מטרת הפרויקט, חישוב הספק של רוכב ללא שימוש במד הספק!

סיכום

בחרתי לחקור את נושא ההספק בהקשר של רוכבי אופניים מכמה סיבות:

1. אני רוכב כביש בעצמי, אחת השאלות שאני שואל את עצמי בכל רכיבה היא מה ההספק שלי? כמה התאמצתי? האם הייתי ביום טוב או רע היום? לאחר שחקרתי קצת את הנושא והגעתי למסקנה שמד הספק זה לא משהו שביכולתי לקנות. אז חשבתי על דרך יצירתית לפתור את הבעיה שלי ושל עוד מיליוני רוכבים בעולם שמחכים בקוצר רוח למד הספק זול.
2. הנושא של הספק בהקשר של רוכבי אופניים הוא נושא שנכנס לתחום הרכיבה רק בשנים האחרונות, ולכן חוקרים אותו רבות, על כן היה לי ברור שאני אהיה הראשון או בין הראשונים שיגיע למסקנות בנושא חישוב ההספק ומציאתו ללא מד הספק.
3. הכלים שקיבלתי משלמה (המורה שלי למדע חישובי) לביצוע חישובים באמצעות המחשב בכלל ושפת הפייתון בפרט התאימו למדי לנושא הפרויקט. השימוש במחשב ללא ספק הקל עלי ואני לא חושב שהייתי מסוגל לבצע ולו עשירית מהפרויקט ללא מחשב.

מהפרויקט "חישוב הספק עבור רוכבי אופניים" למדתי רבות על נושאים הקשורים לפיזיקה, אופניים ואף מדע חישובי. הכרתי את גורמי ההתנגדות לאופניים, והבנתי כיצד הם משפיעים על הספק הרוכב, הכרתי את נוסחת ההספק ואת הקשר שלה למערכת נעה, למדתי להשתמש במטריצות לצורך חישובים (חישוב k1 ו-k2 במקרה זה) והכרתי אפשרויות חדשות בשפת הפייתון. הפרויקט אתגר אותי מבחינה מחשבתית ויישומית, במהלך הכנתו נתקלתי בנקודות שפל ובנקודות שיא (שבראשן עומדת ההשוואה הראשונה בין גרף ההספק הנמדד לגרף ההספק המחושב), עבדתי עליו במשך זמן רב ונעזרתי באנשים סביבי.
נעזרתי במורה שלי, שלמה, באבא שלי (פיזיקאי ורוכב) וברוכב בעל מד הספק שביקש ממני לא לפרסם את שמו וללא נדיבותו הרבה במסירת נתוני ההספק אלי, כל הפרויקט של חישוב הספק עבור רוכבי אופניים לא יכול היה לצאת לפועל.

אני מרוצה למדי מתוצאות הפרויקט, שהן מעל ומעבר לציפיות שהיו לי כשניגשתי לתחילת החקר.
שמחתי לגלות שההשערה ההתחלתית שלי, שקיים קשר בין גורמי ההתנגדות השונים לאופניים להספק הרוכב, נכונה. מבחינתי הגרפים שהתקבלו בסופו של דבר מציגים הצלחה יפה מאוד של הפרויקט והשגת המטרה שלשמה חקרתי את הנושא.

שגיאות שהתקבלו בעת הכנת הפרויקט נובעות ממספר סיבות:

1. אי דיוק במדידת הגובה: מדידת גובה באמצעות GPS לא מדויקת מספיק עבור חישוב הספק שבו יש לדעת שיפוע בכל נקודה בדיוק של סנטימטרים לפחות.
2. שינוי במסת הרוכב: הרוכב שבנתוניו נעזרתי נתן לי הערכת מסה שלו ושל האופניים ליום הרכיבה, הערכה זו לא מדויקת מספיק כלל והדיוק שנחוץ לחישוב מדויק הוא דיוק של גרמים ולא של ק"ג. לצורך

ההסבר, ההבדל במחיר בין אופניים ששוקלים 8 ק"ג לאופניים ששוקלים 7 ק"ג הוא כ-15,000₪, ולא סתם, המשמעות של משקל בהקשר של אופניים היא משמעות גדולה, ולא ניתן להתעלם ממנה. לכן, אחת הסיבות שרוכבים מוציאים מכיסם כסף כה רב על מנת להוריד מספר גרמים היא ההשפעה הגדולה של מסת הרוכב+האופניים על ההספק.

3. מציאת k1 ו-k2 עדכנית לתאריך הרכיבה ה-22/07/2014, והרכיבה הנוספת שאת נתוניה קיבלתי היא מתאריך ה-04/03/2014. על כן קרוב לודאי ש-k1 ו-k2 המתאימים לסוף חודש יולי של שנת 2014 לא מתאימים לתחילת חודש מרץ של שנת 2014 עקב שינוי במסת הרוכב/האופניים ואף שינוי בביגוד המשפיע על האירודינמיות.

4. התעלמות מרוח: עקב מחסור בנתוני מהירות הרוח ברכיבות נאלצתי להתעלם ממנה ולהעריך שאין רוח. ההתעלמות מהרוח הובילה לאי דיוקים רבים וגדולים מכיוון שהשפעתה על הספק הרוכב היא אסטרונומית.

5. שינויים בתנוחת הרוכב: באופני כביש ישנם הרבה מקומות על הכידון בהם יכול הרוכב לאחוז וכתוצאה מכך הוא יכול לרכוב בהרבה תנוחות שונות. בכל תנוחה התנגדות האוויר שונה, ומכיוון ש-k1 (קבוע התנגדות האוויר) נשאר קבוע בכל הרכיבה, הוא לא יתאים לתנוחות מסוימות. שינויים בתנוחה גוררים פעמים רבות שינויים דראסטיים בהספק, עד עשרות ואטים.

בעתיד אנסה לפתור בעיות אלה הגורמות לאי דיוקים, אעשה שימוש במד שיפוע ובמד רוח ואף אנסה למדוד את תנוחת הרוכב. אני מאמין שאם אצליח לשים את ידי על עוד נתוני רכיבות הכוללות הספק נמדד של רוכבים רבים נוספים, אמצא קשר בין סוג האופניים ומבנה גופו של הרוכב ל-k1 ו-k2, ואז אוכל למצוא אותם בלי שימוש במד הספק. כעת, לאחר שראיתי שקיים קשר ממשי בין הגורמים המתנגדים לרוכב לבין הספק הרוכב, אני יודע שיש סיבה טובה להשקיע זמן, כסף ומאמצים בהמשך חקירת הנושא כאשר היעד הסופי שלי הוא פיתוח מד הספק זול ומדויק, שיעשה שימוש בדרך החישובית למציאת ההספק.