**זיהוי שהיות מבוססות מד-תאוצה**

**נכתב ע"י: דניאל בן משה**

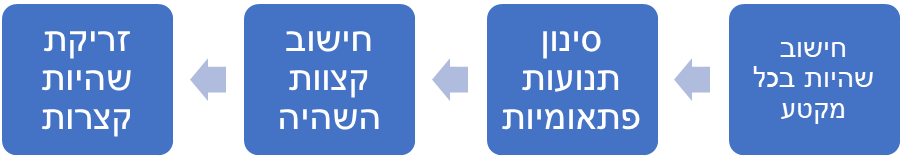
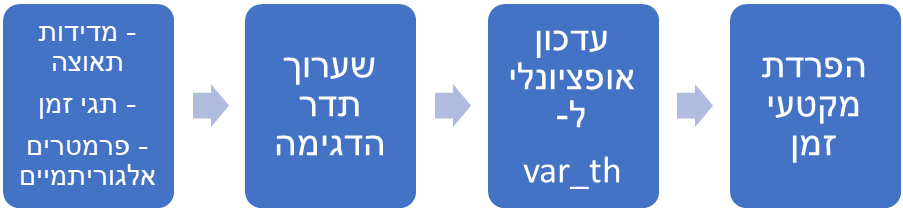
**נכתב בתאריך: 25.3.2020**

**נבדק ע"י: \_\_\_\_\_\_\_\_\_\_\_\_\_**

מטרה:

בהינתן 3 מדידות תאוצה של מכשיר כלשהו ב-3 צירי הייחוס שלו , וכן ותגי הזמן המוצמדים אליהן, מציאת פרקי הזמן בהם המכשיר שהה ולא התבצעה תנועה משמעותית (הליכה, נסיעה...).

תיאור האלגוריתמיקה הכללית:



תיאור משתני הכניסה:

% Inputs:

% accx [Nx1 double] - accelerometer vector along x axis

% accy [Nx1 double] - accelerometer vector along y axis

% accz [Nx1 double] - accelerometer vector along z axis

% timestamp [Nx1 datetime] - time vector corresponding to acc.

% params [1x1 struct] - thresholds etc. regarding algorithm, as

% follows:

%

% win\_size\_sec [double] - sample's environment decision length in

% seconds. Illustration below:

%

% <----- 3 sec ----->

% [ ... s i g n a l ... s i g n a l ...]

% ^

% isStay decision

%

% max\_time\_gap\_pctl [double] - pctile of time gap distribution,

% in range [0 100]. Gaps longer then

% gap corresponding to <max\_time\_gap\_pctl>

% will not take part in "fs" estimation

%

% max\_section\_gap\_minutes

% [double] - maximal time gap between following

% samples, to be considered as same

% section

%

% ecdf\_diff\_th [double] - possible update to var\_th parameter

% by indicating a knee in the ECDF of

% the moving variance of the signal

%

% var\_th [double] - moving variance threshold upon which

% sojourning decision is being made

%

% abrupt\_filt\_time\_const

% [double] - abrupt movement detection filter's

% timeconstant in seconds

%

% abrupt\_pctg\_th [double] - abrupt movement detection threshold.

% Should be between [0 - 1], indicating

% minimal percentage of '1' isStay samples

% around filtered sample. For example:

%

% isStay = [ ... 0 1 1 0 1 1 1 0 0 1 0 0 0 1 0 1 0 1 ... ]

% sample of ^ interest

% | -- window of interest --- |

%

% then sample has '1' at 7/14=50% of its

% environment. samples with more

% than <abrupt\_pctg\_th> (not in

% [%]) will become '1'

%

% min\_stay\_duration\_minutes

% [double] - minimal sojourning duration in minutes.

% shorter detected sojourns are to

% disqualified.

%

%

%

% Outputs:

% isStay [Nx1 bol] - Boolean vector indicating sample sojourning

% stay\_times [Mx2 datetime] - sojourning edges if exist, empty otherwise

% stay\_durations [Mx1 datetime] - sojourning durations if exist, empty otherwise

%

פירוט הבלוקים:

**שערוך תדר הדגימה:**

ראשית, אין הכרח שהדגימות תהיינה מרווחות אחיד בזמן. דגימה לא אחידה מתקבלת כתוצאה מגורמים כגון:

* אי דיוק של השעון במכשיר
* מגבלות של הדוגם (מעגל חומרתי)
* זמן עיבוד והשהיה חומרתית

עם זאת, רוב הדגימות אמורות (ואכן כך) להגיע במרווחי זמן אחידים. לכן נוכל להעריך מהו תדר הדגימה של הסיגנל.

1. ראשית מחושבות קפיצות הזמן בין דגימות עוקבות.
2. לאחר מכן נחשב מהי קפיצת הזמן הקריטית – זו הנמצאת באחוזון max\_time\_gap\_pctl (למשל, אחוזון 60%).
3. עבור כל מרווחי זמן נמוכים מהקפיצה הקריטית, נמצע את ההופכי שלהן לקבלת fs – שערוך תדר הדגימה:

fs = mean(1e3 \* 1 ./ time\_diffs\_msec( time\_diffs\_msec < prctile(time\_diffs\_msec, max\_time\_gap\_pctl) ));

**עדכון אופציונלי ל-var\_th:**

לעיתים מתקבל מקטע תאוצה שרובו מהווה שהייה אך חלק קטן ממנו לא. נוכל לנצל מקרים כאלו על מנת לעדכן את סף ההחלטה הרלוונטי, var\_th. העדכון נעשה באמצעות השלבים הבאים:

1. חישוב הערך המחולט של מדידות התאוצה:
2. חישוב "השונות הנעה" של הסיגנל:

* לכל דגימה נפתח חלון סביבה, באורך של win\_size\_sec (כאשר המעבר בין שניות לדגימות נעשה על כפל ב-fs).
* חישוב השונות המדגמית של כל חלון בגודל N דגימות:
* מעבר לדגימה הבאה בסיגנל
  + דגימות בקצה – החלון נפתח סביב צד אחד בלבד.

למעשה, חישוב זה כולו מתבצע על ידי פונקציה ב-MATLAB:

acc\_movevar = movvar(acc\_abs, win\_size\_smp);

1. חישוב "ההתפלגות המדגמית" של השונות-הנעה:
   1. חישוב היסטוגרמה רבת עמודות (MAX\_HIST\_BINS=1e4) למטרות רזולוצייה, ונרמול ל-pdf ע"י נרמול סכום העמודות ל-1.
2. מציאת עמודת ההיסטוגרמה המאוחרת ביותר (על ציר השונות-הנעה), שבה ערך העמודה נמוך מהסף ecdf\_diff\_th.

ecdf\_knee\_idx = find( mvr\_epdf < params.ecdf\_diff\_th ,1,'last');

knee\_th = hist\_centers(ecdf\_knee\_idx);

1. לבסוף, נעדכן אפוציונלית את הסף var\_th אם הערך שהתקבל נמוך ממנו:

if ~isempty(knee\_th)

var\_th = min( var\_th , knee\_th);

end

**הפרדת מקטעי זמן:**

בחלק זה מוגדרים sections, שהם מקטעי זמן זרים לחישוב השהיה. Section מתקבל כאשר קיימת קפיצת זמן ארוכה מ- max\_section\_gap\_minutes. כאמור, חישוב השהיה יחושב על כל תת-מקטע בנפרד.

**חישוב שהיות בכל מקטע:**

מתבצע על פי השלבים הבאים:

1. חישוב הערך המחולט של הסיגנל (בדומה לנעשה מקודם, רק עבור תת-המקטע בלבד. ניתן לחילופין לקחת את הדגימות הרלוונטיות לתת-המקטע מתוך המשתנה -acc\_abs שחושב קודם.
2. חישוב שונות-נעה לכל ציר בנפרד, וכן עבור הערך המוחלט.
3. מעברי סף:
   1. מעבר השונות הנעה לכל ציר, בסף var\_th/3 מתוך הנחה שמכשיר במנוחה מוחלטת בעל שונות נמוכה יותר – על ידי הקטנת הסף נוכל למנוע התרעות-שווא.

is\_seperate\_axis(curr\_section\_idxs) = all(mvr\_mat < var\_th/Ndims,2);

* 1. מעבר השונות הנעה של הערך המוחלט בסף var\_th.

is\_abs\_stay(curr\_section\_idxs) = mvr\_abs < var\_th ;

1. חיתוך כל התוצאות:

isStay = is\_seperate\_axis & is\_abs\_stay ;

**סינון תזוזות פתאומיות:**

ייתכנו תזוזות רגעיות ופתאומיות אשר יתבטאו בתנודתיות-יתר במד-התאוצה, ובשינויים קצרים מתאימים ב-isStay:

נרצה לסנן תזוזות אלו על ידי פילטר יחידה באורך abrupt\_filt\_time\_const (כמובן, לאחר המרה משניות לדגימות). הפילטר למעשה מממש ממוצע פשוט של כל הדגימות שנכנסות באורך שלו.

לאחר הסינון נקבל וקטור בעל ערכים רציפים בין 0 ל-1 :



לבסוף, כל הדגימות מעל לערך הסף abrupt\_pctg\_th מואלצות להיות 1:

isStay(isStay > params.abrupt\_pctg\_th) = 1;

לטובת החלק הבא, מאולצות דגימות בזמנים שמפרידים sections להיות 0.

**חישוב קצוות השהיה**

החישוב נעשה על ידי מציאת הזמנים שבהם שי שינוי ב-isStay. שינוי חיובי מצביע על תחילת שהיה ושינוי שלילי מצביע על סיומה. יש להתחשב במקרי קצה אופייניים:

* לא זוהתה שהייה כלל

השהייה האחרונה נמשכת עד לסוף המקטע - מוגן אוטומטית על ידי השורה:

isStay(section\_idxs(2:end)) = 0; % force sectioning

**זריקת שהיות קצרות**

שהיות קצרות מערך הסף min\_stay\_duration\_minutes נזרקות.