**מבוא**

במסמך הזה יש כל מה שצריך כדי להריץ ולנתח את המרדף של שפריריות אחרי מטרה נעה. הניסויים נערכו ע"י זיו קסנר במסגרת לימודי דוקטורט במעבדה לביומכניקה בהנחייתו של דוקטור גל ריבק באוניברסיטת תל אביב.

בתחילת המסמך יש הוראות הפעלה לביצוע הסימולציות והסברים כלליים מאוד על תפקידה של כל פונקציה, לאחר מכן יש הסבר קצר על ניווט פרופורציונלי (לחילופין, "ניווט בהגבר קבוע"), ובסופו יש רשימה מסודרת של הפונקציות בהן השתמשתי. בנוסף, בתוך גופן של רוב הפונקציות יש הסברים מפורטים יותר.

בכל מקרה – אני זמין ב 0507355900 או ב [yoavbdavid@gmail.com](mailto:yoavbdavid@gmail.com).

**איך להפעיל את הסימולציה?**

1. לפתוח את קובץ main: בין השורות 6-46 יש לבחור את סוג הניסוי (מטרה נייחת, מטרה מתמרנת ב 1Hz, 2Hz, 2.5Hz, או מטרה באמפליטודה כפולה) ולהוציא אותו מהערה.

שים לב: יש בכל אחת מהתדירויות

1. כדי לשאוב את המידע הגולמי של הניסוי (מיקומי השפרירית והמטרה בכל אלפית שנייה), עליך לשנות בשורה 55 את שם המשתנה textFileName לקידומת של מיקום הקבצים במחשב עליו את\ה עובד\ת.
2. לאחר מכן, פונקציית ה main תשאב את המידע ותנתח אותו: תחשב מיקומים, מהירויות, תאוצות, זויות ועוד. הסברים מפורטים בגוף הפונקציה. ה main מחשב גם גדלים שלא נעשה בהם בגרסה הנוכחית.
3. עד כאן השתמשנו ב main. כעת יש לפתוח את הסקריפט k\_trials\_with\_tau2. התפקיד שלו הוא לרוץ על כל הניסויים שבוצעו באמפליטודה מסוימת, ולכל אחד מהם להריץ סימולציית ניווט בהגבר קבוע (הסבר על האלגוריתם הזה – בהמשך).

בתחילתו (שורות 5-6) בוחרים ערכי K ו tau לנסיונות (tau אופציונלי: בעזרת שיטה זו לא הצלחתי למצוא ערכי tau הגיוניים. הסבר בהמשך מה ניסיתי לעשות כדי להתגבר על זה).

עבור כל ניסוי, הסקריפט מתחיל במיקום ובזווית הראייה של המעוף האמיתי של השפרירית. הסקריפט מנסה הרבה מאוד ערכי K (אפשרי גם לסמלץ ערכי tau, קבוע זמן התגובה), ועבור כל ערך שכזה הוא בונה מסלול לעבר המטרה. לאחר מכן, השגיאה בכל צעד זמן נמדדת ולאחר תום המסלול מחושבת שגיאה ממוצעת לאורך המעוף. כך, עבור כל ניסוי, ועבור כל ערך K, מחושבת השגיאה: והסקריפט בוחר את ערך K (ו tau אם רוצים) שנותן את השגיאה הקטנה ביותר. לדוגמה, במרדף שבו נבחנו עשרה ערכי K ועשרה ערכי tau, תיבנה הטבלה הבאה עבור כל צעד:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K=10 |  |  |  |  |  |  | … | K=2 | K=1 |  |
|  |  |  |  |  |  |  |  |  |  | Tau = 0.02 |
|  |  |  |  |  |  |  |  |  |  | 0.03 |
|  |  |  |  |  |  |  |  |  |  | … |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | .. |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

בכל תא בטבלה יהיה ערך סטייה בין המיקום הנוכחי על המסלול המקורי שנמדד לבין המיקום הנוכחי על המסלול שמסומלץ עבור ערך K ו tau הנוכחי. כאמור, טבלה כזו תיבנה בכל צעד זמן, ולבסוף הממוצע של כל השגיאות יחושב ואז ייבחרו K ו tau שייתנו את השגיאה הממוצעת הקטנה ביותר.

1. כעת עוברים ל plot\_best\_k. הסקריפט הזה בונה מסלולים רק עבור ערכי ה K שנבחרו כטובים ביותר ומחזיק את הקואורדינטות שלהם. ניתן להוציא מהערה את השורות האחרונות בתוך הלולאה החיצונית ולקבל גרף שמציג את מסלול המטרה, מסלול הטיסה שנמדד ומסלול הטיסה שסומלץ על ידי ערך K המיטבי.
2. כעת, בהנחה שלא בחרנו לסמלץ ערך tau, עלינו למצוא את ערכו בדרך אחרת כשבידינו רק ערך K. לשם כך היו מספר רעיונות:

* שימוש ב Cross correlation: למציאת הזזה כמה שיותר טובה בין האות שנקבע בסימולציה עם זמן תגובה 0tau= לבין הניסוי האמיתי. לדוגמה, אנחנו יודעים את זווית הראייה המדודה מחישוב ב main, ואת זוית הראייה בסימולציה אנחנו יודעים מ plot\_best\_k. כך נוכל למצוא הזזה ביניהם.

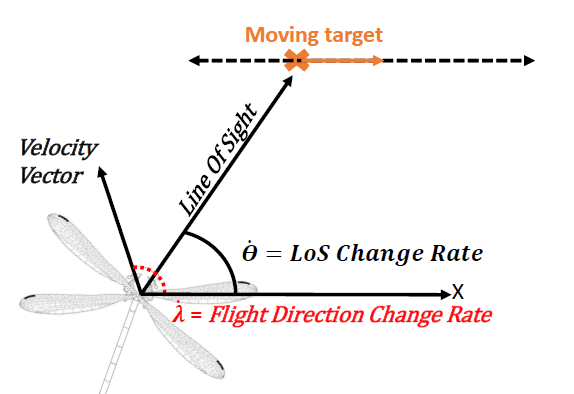
הבעיה: קפיצות זווית בין 180 ל180- שהורסות את החישוב. ניסיתי לכתוב פונקציות שמבטלות את הקפיצות האלה או מתעלמות מהן (limiting, no\_jumping) אבל זה לא ממש הלך.

* ניסיון לעבוד עם ההפרש בין המדידה לסימולציה: אם שוב ניקח לדוגמה את זוית הראייה המדודה לעומת המסומלצת, כדי להתגבר על המכשול של הקפיצות ניסיתי לעבוד על חישוב ההפרש ביניהם. את ההפרש חישבתי איבר-איבר במערכים הנבדקים (זווית ראייה), וכל פעם הזזתי את הסימולציה צעד אחד קדימה. ציפיתי שבהזזה "נכונה" (כלומר הזזה שהיא זמן התגובה של השפרירית) אני אראה שממוצע ההפרש בין הוקטורים הכי קרוב ל0. גם זה לא עבד מספיק טוב. קיים ב trial.

אני חושב שאחת מהדרכים האלה יכולות לעבוד, אולי לא כתבתי אותן מספיק טוב.

**הסבר על ניווט פרופורציונלי**

בבסיס האלגוריתם עומדת השאלה כיצד נקבע מסלול של רודף (במקרה שלנו – שפרירית) לעבר מטרה.

אלגוריתם זה מציע את השיטה הבאה: קצב שינוי הכיוון של הרודף מתבצע ביחס פרופורציונלי קבוע למהירות הזויתית של המטרה ביחס למיקום הרודף.

משוואה זו מחושבת בכל צעד מחדש, כאשר Lambda מייצגת את כיוון וקטור המהירות ביחס למרחב ו Theta את זוית קו הראיה (יש לשים לב שבשני אגפי המשוואה מדובר על **קצב** **שינוי** הכיוון):

]

כפי שנאמר בחלק ההסבר על הרצת הסימולציות: אנו בוחנים ערכי K שונים ובוחרים לבסוף את הערך בעל השגיאה הממוצעת הקטנה ביותר.

במשוואה מעלה מובלעת הנחה לא מדויקת ששינוי הכיוון מתבצע מיידית עם המדידה. בפועל אנחנו יודעים שיש עיכוב של מערכת העצבים ושל יכולת התנועה של השפרירית (הגוף שלה בעל תנע ולא משנה את כיוון המעוף ברגע משק כנפיים). כדי לקחת את העיכוב הזה בחשבון, אנחנו משנים את צורת המשוואה לצורה זו:

כאשר tau מייצגת את זמן התגובה הכולל (מערכת עצבים + יכולת פיזית).

פונקציות: