

# רובוטים ניידים – תשפ"ו

## תרגיל 1 (עקיבה אחר קו - תנועה מבוקרת משוב, בקר PID, חישוב מסלול בעזרת אודומטריה)

בתרגיל זה תבנו רובוט בעל הנעה דיפרנציאלית, המבצע משימת עקיבה אחר קו כהה על רקע בהיר, באמצעות חיישן בהירות יחיד. הרובוט ינוע מהר ככל האפשר על פני המסלול ויחשב את מיקומו ביחס לנקודת התחלה על פי מדידת סיבובי הגלגלים. המיקומים ישמרו לקובץ שיעובד ויוצג offline בעזרת תוכנת מחשב שתכתבו.

התרגיל מחולק לשני סוגי משימות. **הסוג הראשון** הן בדיקות במעבדה של ביצועי הרובוט שבניתם. **הסוג השני** הן משימות להגשה בהן תציגו את תהליך הפיתוח ואת עיבוד הנתונים. **ישנם גם סעיפי בונוס.**

### משימת עקיבה ראשונה

1. **(בדיקה במעבדה)** על הרובוט שבניתם להצליח לנוע במשך שני סיבובים ברצף על המסלול הסגור והקמור שבמעבדה תוך כדי עקיבה אחר הקו הכהה. אם הרובוט מאבד את הקו עליו למצוא אותו ולהשלים את שני הסיבובים.

הנחיות והנחות:

- א. הרובוט מתחיל את משימתו כשהוא מונח על הקו הכהה על ידי המפעיל.
- ב. הרובוט לא חייב להצליח לעקוב לשני הכיוונים, אולם אם הוא מונח בכיוון שאינו מועדף, עליו להצליח לתמרן אל הכיוון המועדף ומשם להשלים שני סיבובים.
- ג. על הרובוט לסיים את המשימה בזמן מהיר ככל האפשר. **ינתן בונוס לפי הישגים ביחס לביצועי הרובוטים של הקבוצות האחרות.**
- ד. סיום המשימה יהיה כאשר הרובוט יתקל במכשול אשר נניח לפניו בקו הסיום, ממש לפני הגעתו.
- ה. המפעיל ימדוד את הזמן מרגע התחלת המשימה ועד השלמת שני סיבובים. המדידה תתבצע פעמיים: פעם אחת כשכיוון ההתחלה מתאים לסיבוב בכיוון השעון ופעם שנייה עבור כיוון ההתחלה ההפוך.
- ו. הערכו לצילום סרטון מלא של המשימה.

## 2. (הגשה)

- א. הסבירו בפרוט את השיטה בה הרובוט שלכם מבצע את משימה 1. התייחסו לשיקולים בבחירות שבחרתם, לחלוקה לשלבים שונים של המשימה, לכיול חיישן הבהירות וקביעת מיקומו, לדרך בה הגברתם את אמינות השיטה שלכם, לדרך בה קיצרתם את הזמן שלוקח לרובוט להשלים את המשימה ול tradeoff בין אמינות ומהירות (אם היה כזה). צרפו צילומים של הרובוט שלכם לתיעוד הגרסאות השונות (עד לגרסה הסופית) + הסבר השינויים.
- ב. הציגו את תוצאות מדידות הזמנים שערכתם עבור שלבי הפיתוח השונים.
- ג. צרפו את הקוד הסופי למימוש זה.

## משימת עקיבה שנייה

3. (בדיקה במעבדה) השתמשו בבקר PID (proportional–integral–derivative) לביצוע משימת העקיבה אחר הקו. בקר מסוג זה מנסה להביא משתנה של המערכת אל ערך יעד מסוים. ההפרש הרגעי בין המשתנה לערך היעד זו השגיאה עליה השיטה מתבססת. בד"כ השגיאה רציפה ובעלת כיוון, כלומר יש משמעות גם לערכה של השגיאה וגם לכך שהשגיאה חיובית או שלילית. עליכם למצוא דרך להתאים את השימוש בבקר PID לבעיה של עקיבה אחרי הקו בעזרת רובוט הלגו שלכם. שימו לב **שאותו רובוט** צריך לבצע את משימה 1 ואת משימה 3, כשהשוני הוא בתוכנה בלבד.

בחנו את ביצועי הרובוט על אותה זירה ובאותה דרך כמתואר בסעיף 1.

## 4. (הגשה)

- א. הסבירו את המימוש שלכם לבקר. התמקדו בבחירת המשתנה המבוקר וכיצד מדדתם אותו. תארו כיצד כיילתם את המקדמים של הבקר.
- ב. הציגו את תוצאות מדידות הזמנים שערכתם עבור מימוש PID של הרובוט העוקב. נתחו את התוצאות בהשוואה למימוש שהצגתם בסעיף 1 ו 2.
- ג. צרפו את הקוד הסופי למימוש זה.

## חישוב המסלול

המסלול יחושב בעזרת אודומטריה: השתמשו במידע המתקבל מחיישני הסיבוב כדי לחשב את מסלול התנועה של הרובוט.

הדרכה: עליכם לחשב את כיוון התנועה ואת מידת ההתקדמות בכיוון זה. היעזרו בחישוב של רדיוס הסיבוב של הרובוט כתלות במידת הסיבוב של הגלגלים. תוכלו להשתמש בהיקף הגלגלים כיחידת

האורך של המסלול הנוצר ואחר כך להמירה לס"מ. כדאי לבדוק את קוד האודומטריה שלכם עבור תנועות ידועות מראש עוד לפני ביצוע משימת העקיבה אחר הקו.

5. (בדיקה במעבדה) משימת חישוב המסלול על ידי אודומטריה תיבדק לשני הכיוונים ובשני אלגוריתמי העקיבה שבניתם. על כן הפלט למשימה זו הוא 4 מסלולים.

6. (הגשה)

- א. הסבירו במפורש את דרך החישוב בה השתמשתם.
- ב. הציגו וויזואלית את ארבעת המסלולים שעשה הרובוט שלכם בסעיף 5. במהלך המשימה עליכם לשמור נתונים לקבצים שאחר כך ניתן להוריד למחשב ולהציג.
- שימו לב לפרטים העדינים של המימוש: האם להשתמש באותה גרסה בדיוק של קוד עבור משימת העקיבה המהירה ועבור משימת מציאת המסלול? יכולות להיווצר בעיות הקשורות למשך הזמן הנוסף הנדרש לביצוע השמירה לקובץ והשפעתו על מהירות ביצוע לולאת הבקרה. אילו משתנים בדיוק כדאי לשמור? כל כמה זמן לשמור אותם? מה גודל הקובץ שאתם מייצרים?
- ג. (בונוס) הציגו את המסלול כפלט גרפי על מסך הרובוט. הגישו צילומי מסך.
- ד. (בונוס) חשבו והציגו גם את צורת הקו שאחריו עקב הרובוט (כיצד?).
- ה. צרפו את כל הקוד הסופי למימוש זה – קוד ה `NXC \ Lejos` הרץ על הרובוט, וכל קוד אחר הרץ על המחשב ( `Java / MATLAB / Python / Julia` ).

### הגשה:

1. הבדיקות במעבדה יערכו ביום שישי, ה 28.11.25.
2. הגשה (על ידי אחד מחברי הצוות) בתיבת ההגשה באתר הקורס עד יום שלישי 2.12.25 לפני חצות.
3. מותר להשתמש במקורות חיצוניים (תיעוד, מאמרים, הדגמות, קוד, כלי AI). **חובה לציין כל מקור.**

4. יש לרשום שם ות"ז של כל חברי הצוות בראשית כל קובץ או מסמך.
5. הקפידו על סטנדרטים של כתיבת קוד: ארגון, שמות משמעותיים, תיעוד.
6. אם יש בעיה להעלות קבצים עם סיומות מסוימות, קבצו אותם בתוך קובץ zip אותו תעלו.
7. הגישו גם את קבצי הווידאו של הבדיקות במעבדה או (אם הם גדולים מדי) קישור לתיקיית שיתוף בה נמצאים קבצי הווידאו.

## **מקורות:**

1. הסתכלו היטב בתיעוד של Lejos, במיוחד על DifferentialPilot, OdometryPoseProvider, גם אם אתם מתכנתים בסביבה אחרת.

2. בקר PID

[http://en.wikipedia.org/wiki/PID\\_controller](http://en.wikipedia.org/wiki/PID_controller)

[http://www.inpharmix.com/jps/PID\\_Controller\\_For\\_Lego\\_Mindstorms\\_Robots.html](http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html)

[http://www.seattlerobotics.org/encoder/200108/using\\_a\\_pid.html](http://www.seattlerobotics.org/encoder/200108/using_a_pid.html)

3. אודומטריה

<http://credentiaity2.blogspot.co.il/2010/06/going-from-odometry-to-position-in-two.html>

<http://en.wikipedia.org/wiki/Odometry>

<http://rosum.sourceforge.net/papers/DiffSteer/>

<http://robotics.stackexchange.com/questions/1653/calculate-position-of-differential-drive-robot>