Imperial College London

Animal Locomotion and Bioinspired Robotics

# Target Interception

Peter **GUAN** – 02031053
Human and Biological Robotics

Department of Bio-engineering
Imperial College London
Date: March 29, 2022

# Contents

# 1 Introduction

# 2 Methodology

## 2.1 Pursuit Updating Mechanism

PID framework is descried as follows,

$$\dot{\theta}_h = K_p(\theta_r - \theta_h) + K_I \int (\theta_r - \theta_h) + K_D \dot{\theta}_r, \tag{2.1}$$

where $\theta$ is the angle in polar coordinate, subscript $h$ and $r$ represent head vector of the agent and target, respectively, and coefficients $K_p$, $K_I$ and $K_D$ are the corresponding coefficients in PID controller. In addition, the angel $\theta_r - \theta_h$ can be calculated by,

$$\theta_e = \theta_r - \theta_h = \mathrm{sign}(\cos^{-1}(\boldsymbol{v_r} \cdot \boldsymbol{v_h})), \tag{2.2}$$

where $\boldsymbol{v}$ is the normalised vector of the corresponding object, and sign function is determined by,

$$\mathrm{sign}(\cos^{-1}(\boldsymbol{v_1} \cdot \boldsymbol{v_2})) = \begin{cases} \cos^{-1}(\boldsymbol{v_1} \cdot \boldsymbol{v_2}) & \boldsymbol{v_1} \times \boldsymbol{v_2} \geq 0 \\ -\cos^{-1}(\boldsymbol{v_1} \cdot \boldsymbol{v_2}) & \boldsymbol{v_1} \times \boldsymbol{v_2} < 0 \end{cases} \tag{2.3}$$

Meanwhile, $\dot{\theta}_r$ can be derived by,

$$\dot{\theta}_r = \frac{\theta_r^{(i)} - \theta_r^{(i-1)}}{\delta t} = \frac{\mathrm{sign}(\cos^{-1}(\boldsymbol{v_r^{(i)}} \cdot \boldsymbol{v_r^{(i-1)}}))}{\delta t}, \tag{2.4}$$

where $\delta t$ is the time step that is set to 0.01 second in this project, and the superscript $i-1$ and $i$ mean the respective previous and current steps. Then a new head vector angle can be evolved by using $\dot{\theta}_h \cdot \delta t$. To make the agent turn with a certain angle generated by PID controller, a rotation matrix [1] should be used to evolve the $(i+1)$-th head vector from $i$-th head vector, which is describe by,

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \; \boldsymbol{v_h^{(i+1)}} = R(\dot{\theta}_h \cdot \delta t) \times \boldsymbol{v_h^{(i)}}, \tag{2.5}$$

where $\theta$ is the rotation angle. Thus the new position $\boldsymbol{x_a^{(i+1)}}$ of target can be derived by,

$$\boldsymbol{x_a^{(i+1)}} = \boldsymbol{x_a^{(i)}} + |\boldsymbol{v_h}| \cdot \delta t \cdot \boldsymbol{v_h}, \tag{2.6}$$

Meanwhile, to tackle the problem that the target is invisible for the agent because the field of view of the camera is 50°, we employ that the agent should rotate the head vector by $\pi/6$, where the sign should be determined by negating the Equation (2.3). Any details can be referred to the agent python file in Target Interception.

## 2.2 Evaluation Matrices

# 3 Results and Discussion

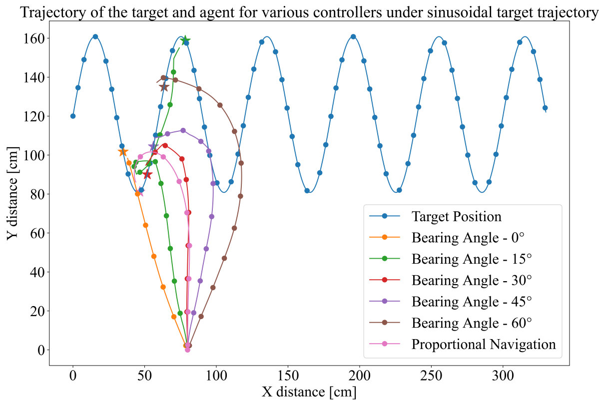
## 3.1 Simple Pursuit and Constant Bearing

PI controller was simulated for simple pursuit and constant bearing. To check the PI controller performance with various bearing angle, a series of bearing angles (i.e., $[0°, 15°, 30°, 45°, 60°]$, where the bearing angle of 0° is the same as simple pursuit) are simulated in the light of the pursuit updating mechanism introduced in Section 2.1. And the chasing trajectories are shown in Fig.3.1a and 3.1c. For the sinusoidal prey route, it appears that the smaller the bearing angle, the faster the agent will intercept the target (Fig.3.1b). Theoretically, the agent would fail to follow the target for the large bearing angle such as 60°, which is because the field of view of the camera is 50°. However, the success of interception as shown in Fig.3.1a is due to the mechanism of rotating a certain angle once the agent cannot see the prey. Meanwhile, Fig.3.1b shows that the best pursuit is when the bearing angle at 0° (i.e., the simple pursuit). In addition, Fig.3.2 introduces the frequency of error angles during the bot chasing the target given various bearing angels and proportional navigation. It can found that the error angle frequency focus on the area near the bearing angle, which is consistent with the fact of the bearing angle (i.e., the agent is trying to keep the constant view angle referred to the target).

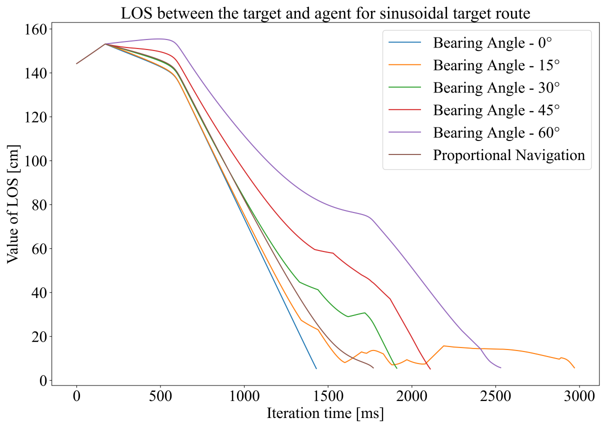
Move on to linear prey route, the agent shows the similar success of intercepting the target. And the successful controllers take close time to get the target except for bearing angle at 15°, which is because small bearing angle needs longer time to keep the bearing angle for linear trajectory. However, simple pursuit and bearing angle at 30° fail to follow the target, the reason of the later failure might be that rotating a certain angle mechanism interrupts its evolution. And ideally, the simple pursuit will not follow the target since they need to be parallel.

## 3.2 Impact of Delay

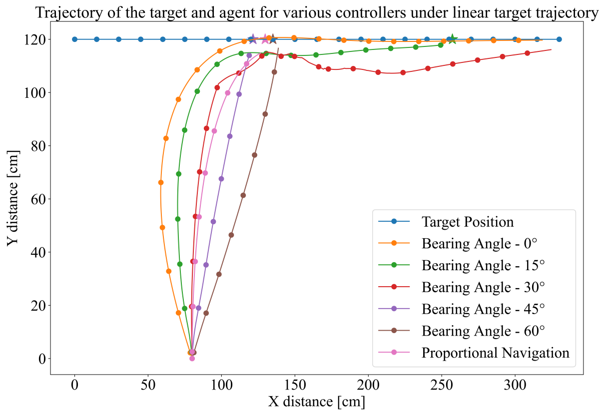
# 4 Conclusion



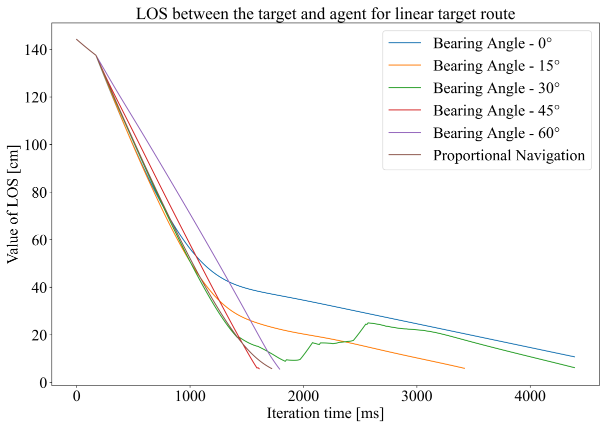

(a) Trajectory for sinusoidal route.




(b) LOS changes for sinusoidal route.




(c) Trajectory for linear route.




(d) LOS changes for linear route.

**Fig.** 3.1: Bearing angle and proportional navigation performance comparison. (a) and (c) give the trajectory comparison for sinusoidal and linear prey routes, respectively, where the coloured star markers represent the interception points of the corresponding coloured controllers. Meanwhile, (b) and (d) demonstrate the LOS distance for different controllers from the starting point to the time they intercept the target.

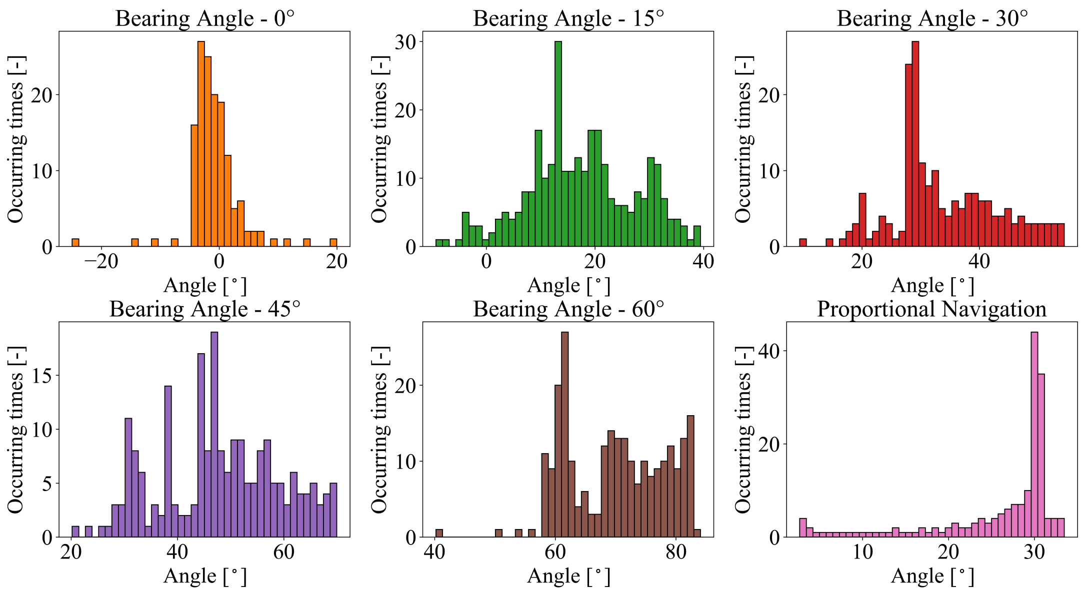



**Fig.** 3.2: Error histogram distribution for various bearing angles and proportional navigation for sinusoidal target route.

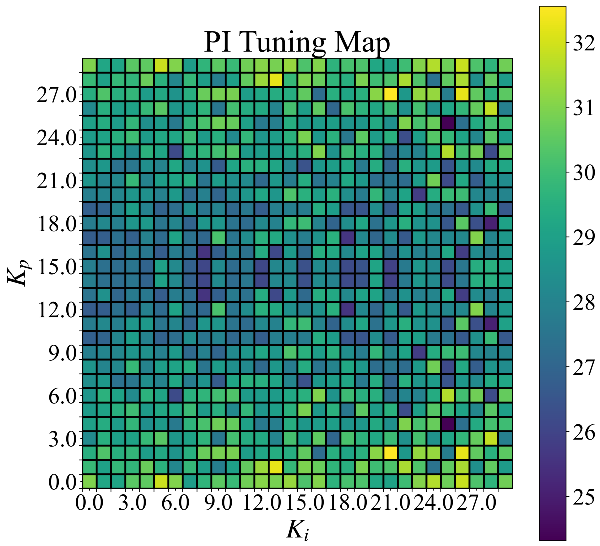



**Fig.** 3.3: PI tuning map. The minimum score 24.32 lies on two points, i.e., $K_p = 4, K_i = 25$ and $K_p = 25, K_i = 25$.

# References

[1] Kevin S. Galloway and Biswadip Dey. Constant bearing pursuit on branching graphs. *2017 IEEE 56th Annual Conference on Decision and Control, CDC 2017*, 2018-January(Cdc):4410–4415, 2018.