

From the formula that

If is too small, then the whole constrain will not be such sensitive, which means that only if robot is much closer to the obstacles, the CBF function can come into play.

From the figure that the more is, the trajectory has more distance from the obstacle, which means that the forward-invariant set is smaller and more conservative. When is 1, the robot cannot adjust its brake to drive itself away and it will go through it, which can be the problem result in robot crash. As to the forward-invariant set, it will contain more the chosen of robot’s trajectory which are high risky and closer to the border of obstacles.



文本, 信件

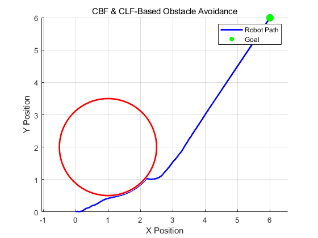
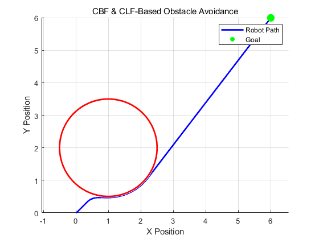
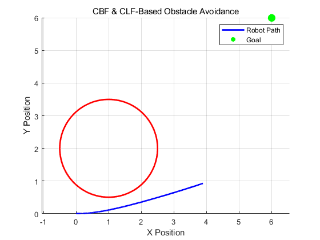
AI 生成的内容可能不正确。

More penalty p will lead the robot trajectory more stick to the obstacles shape.

Gama 1 will affect the speed of convergence.

K will determine whether the range robot can go is big or small.

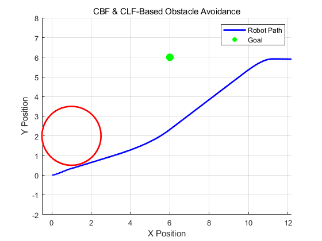
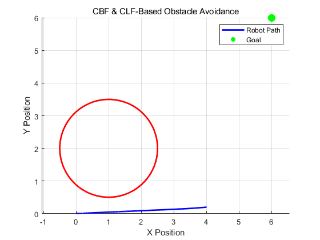
Penalty parameter is to make the system weigh the Control Lyapunov Function (CLF)and safety constraint (CBF). In other words, it determines how much the system needs to satisfy the soft constraint. When is 0.01, the system satisfies the hard constraint by sacrificing CLF, which causes the system to be too slack to reach the target position even though the system is in a safe distance. As equals to 10, the system finds the balance between the constraint. After bypassing the obstacle, the robot still moved a little bit around the obstacle, making the path gentler and the speed more stable. Continue to increase to 5000, it put more emphasis on the CLF, which result in that it modifies the path after bypass the obstacle and finally goes straightly to the target. However, too much large can leads no solution to the system, the result of =8000 is shown in figure. In conclusion, the less is, the more system will slack on the CLF constrain.

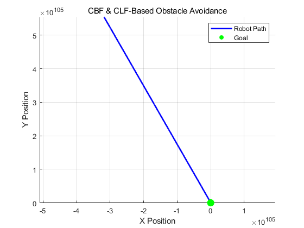
图表, 折线图

AI 生成的内容可能不正确。

*Figure X trajectory of robot when*

CLF is a stable function which can lead the robot finally get to target and determine the convergence speed of this function. In other words, mainly used to adjust the speed and acceleration of the robot. From the Figure, when , robot is very conservative and almost only wanders around the edge of obstacle’s safe range rather than target. As the grows up to 0.1, it can make robot towards the right direction, but the speed of convergence is not enough, which means it still to be increased. As the , the robot starts directly from the initial point and bypasses the obstacle, then quickly approaches the target point. It is more efficient, and path is shorter than the previous trajectory. Finally, when is too large, the robot is uncontrollable, trajectory is extremely far from the point and obstacle. The reason for this phenomenon is that it is so aggressive for the CLF, which forces system to convergence in an unrealizable speed and controller output an extreme input to the robot. Hence, setting up an appropriate value of can make system have good performance on driving operation.

图表, 折线图

AI 生成的内容可能不正确。

*Figure X trajectory of robot when*