Test method

First a test method is designed to test the different controllers. A test consists of 8 different trails, during each trail Bob will be initialized at $p = (x, y, \theta) = (0, 0, \frac{\pi}{2})$ and has to move towards one of the configurations shown in figure 1. The goal configurations are equally spaced on a circle with a radius of 1.5 meter. The goal angle is always pointing towards the origin.

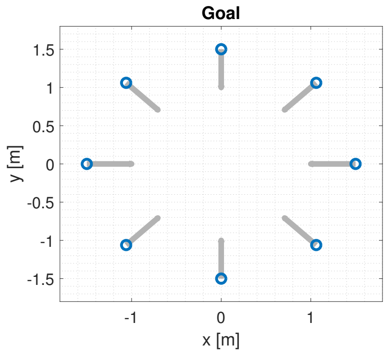


Figure 1: Goal configurations during one test run.

Closed loop Control

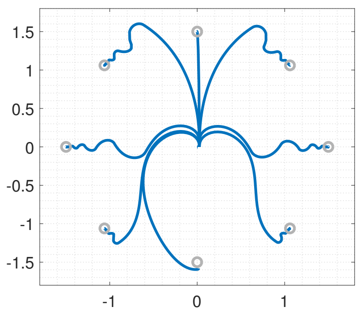
The following controller is implemented:

$$v = k_\rho \rho$$
$$\omega = k_\alpha \alpha + k_\beta \beta$$

Multiple tests were peformed for different controller parameters mentioned in table 1, where also the mean completion time is shown. The trajectories can be seen in figure 2.

Table 1: controller parameters, ans results

	case 1	case 2	case 3	case 4
k_ρ	0.5	1	0.5	0.5
k_α	1.5	1.5	3	1.5
k_β	-0.6	-0.6	-0.6	-1.2
$\bar{t}[s]$	20	11	27	69



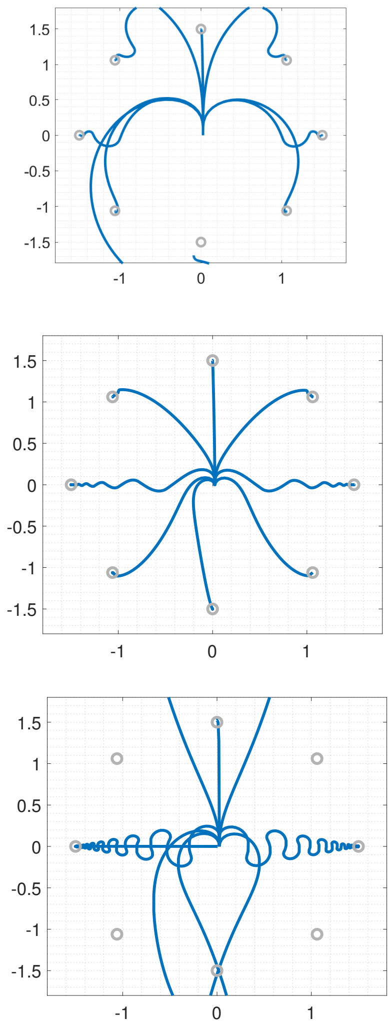


Figure 2: Paths followed by Bob for 4 different cases, ascending top to bottom. On the x-axis the x position is shown, on the y-axis the y position

In case 1 the controller seems fairly balanced, although the k_β seems to be a little high compared to k_α causing Bob to make a large detour. In case 2 the velocity is clearly too high resulting in a large radius of curvature in the bends, we can see a large overshoot at the goal $(0, -1.5[m], \pi/2)$. The smoothest curve has been achieved in case 3, with a fairly high k_α. However the completion time was long since the high k_α is pushing Bob away from the goal orientation. In the last case the large k_β prevents Bob from reaching its goal

Bi-directional

We can see from figure 2 Bob first has to make a large turn to reach the goal configurations underneath the line $y = 0[m]$. A bi-directional controller is implemented which allows Bob to directly move backwards without having to make a large turn. The controller works as follows:

1. Set direction constant: dir $= 1$
2. Check if bi-directional controll is set to true
   a) Check if $|\alpha| > \pi/2$
      i. Set: dir $= -1$
      ii. Adjust: $\alpha = \mathrm{normAngle}(\alpha + \pi)$;
3. Compute: $v = \mathrm{dir} \cdot k_\rho \rho$;
4. Compute: $\omega = k_\alpha \alpha + k_\beta \beta$

Simply put if the goal configuration is located behind Bob it will move backwards.

Constant Velocity

During the previous experiment the velocity of Bob decreased while he was nearing its target, which leads to very slow motion towards the end of the trajectory. A potential fix would be giving Bob a constant velocity, but adjusting the radial velocity to keep the trajectory the same. In other words we want to design a controller with v_{2_c} and ω_2 for which:

$$\frac{v_1}{\omega_1} = \frac{v_{2_c}}{\omega_2} = r, \qquad \rightarrow \qquad \omega_2 = \omega_1 \frac{v_{2_c}}{v_1}$$

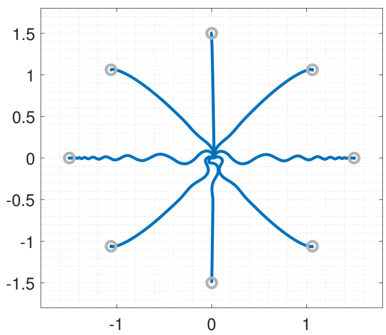
Where v_1 and ω_1 are from the original controller. Note that if $v_2 >> v_1$ the angular velocity becomes very high, this might cause instability.

Results and Discussion

tests were run with $k_\rho = 0.5$, $k_\alpha = 3$, $k_\beta = -0.6$, and $v_{2_c} = 0.3[m/s]$. Again 4 different cases have been tested discussed in table 2.

Table 2: controller parameters, ans results

	case 1	case 2	case 3	case 4
bi-dir	false	true	false	true
const$_v$	false	false	true	true
$\bar{t}[s]$	25	23	7	6



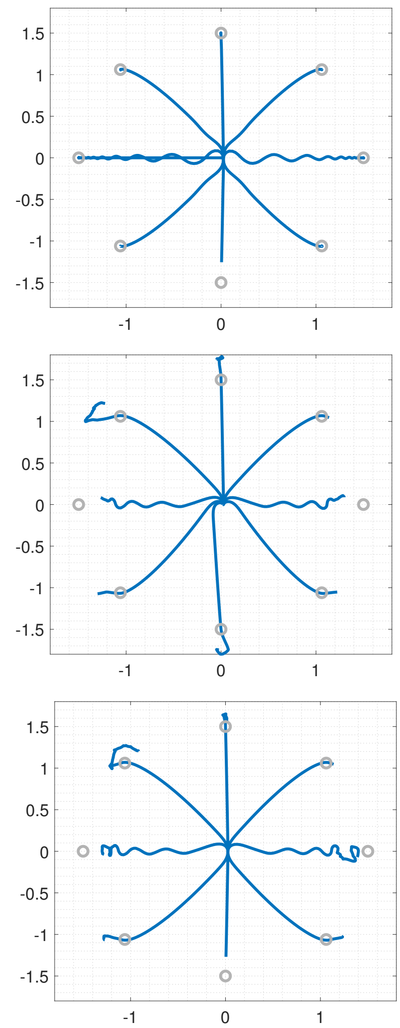


Figure 3: Paths followed by Bob for 4 different cases, ascending top to bottom. On the x-axis the x position is shown, on the y-axis the y position

With the use of the bi-dir controller Bob does not have to make sharp bends anymore, making its path more elegant. The const$_v$ controller strongly reduces the time for Bob to meet its goal. However when Bob approaches the goal the ω setpoints become very large, Bob is not able to reach these high angular velocities and will start spinning, resulting in a very sloppy path. combining the bi-directional and const$_v$ controller does not significantly reduce the goal time since it takes the sharp bends very quickly. We can conclude that the designed controller works well when there is a distance between Bob and it's goal, however when Bob gets close to its target the controller becomes unstable. Since in the test method used Bob had to approach certain non-moving targets its behaviour seems insufficient, Bob performs much better when the target is held a certain distance in front of Bob.