Assignment 1

SC42090 Robot Motion Planning and Control

Xander Guy Gerrmann
4139186

March 3, 2018

Exercise 1.2

Figure 1 shows the resulting paths for different sets of parameters. The parameters are varied one by one, where the rest of the parameters is set to their default values: $\alpha = 1.5$, $\beta = -0.7$, $\rho = 0.5$. The parameters are chosen such that $0 < k_\alpha + \frac{5}{3}k_\beta - \frac{2}{\pi}k_\rho$ still holds. As can be seen, higher α lead to a trajectory that goes more directly towards the final position, resulting in a shorter path. Increasing β causes the robot to approach the target pose with an increasingly correct orientation, resulting in less rotation at the final position. Increasing ρ determines the forward velocity proportional to the distance to the target position. Low ρ results in more dominant orientation control initially compared to a higher initial forward velocity for for higher ρ.

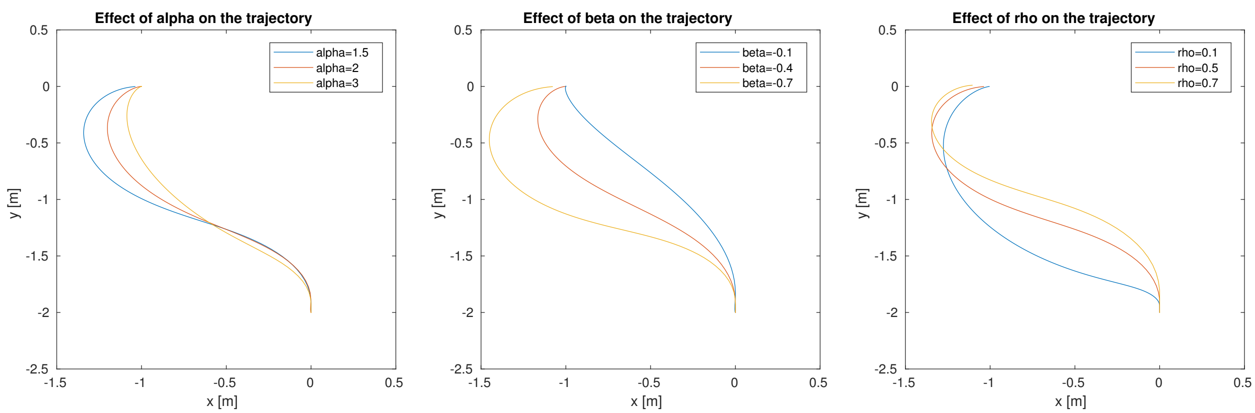

Figure 1: Path comparison of different values for α, β and ρ. Legend correction: $\alpha = k_\alpha$, $\beta = k_\beta$, $\rho = k_\rho$.

Exercise 1.3

Computation of ˇu and ω

Since the path does not change when the ratio v_u/ω remains constant. Therefore, if we scale both v_u and ω with the same factor c, the path should not change. The scalar c is determined by the ration between the calculated v_u and the desired velocity v_{des}.

$$c = v_{des}/v_u \tag{1}$$
$$v_u^* = v_u c \tag{2}$$
$$\omega^* = \omega c \tag{3}$$

For determining the wheel velocities, v_u and ω^* are used. To allow for backward motion, we set a condition on α. When $|\alpha| > 0.5\pi$, the calculated v_u^* is set to a negative value, $v_u^{**} = -v_u^*$. Then v_u^{**} is used for determining the wheel velocities (together with ω^*).

Description and evaluation of tests

For the controller validation I have performed two tests (excluding the results from the previous parameter comparison).

1. Positioning the target on the backside of the robot while the robot is facing the wall
2. Trajectory comparison for the scaling of the forward velocity

For testing the backward motion, a new scene has been created (see Figure 2, top-left). When no backward velocity is allowed, the robot gets stuck in the corner. When a backward velocity is allowed, the robot drives backward, while turning then continues turning while driving forward again. See the top-right figure in Figure 2 for the comparison of the trajectories. For testing the velocity scaling implementation, the same setup is run with velocity scaling enabled and disabled. The results are shown in the bottom-left figure in Figure 2. Also the velocity v_u^* for each simulation step is plotted in the bottom-right figure of Figure 2. The trajectories are practically equal, and the velocity is constant. Also when the velocity is not set to be constant, one can see it decreasing over time (as the robot approaches its target).

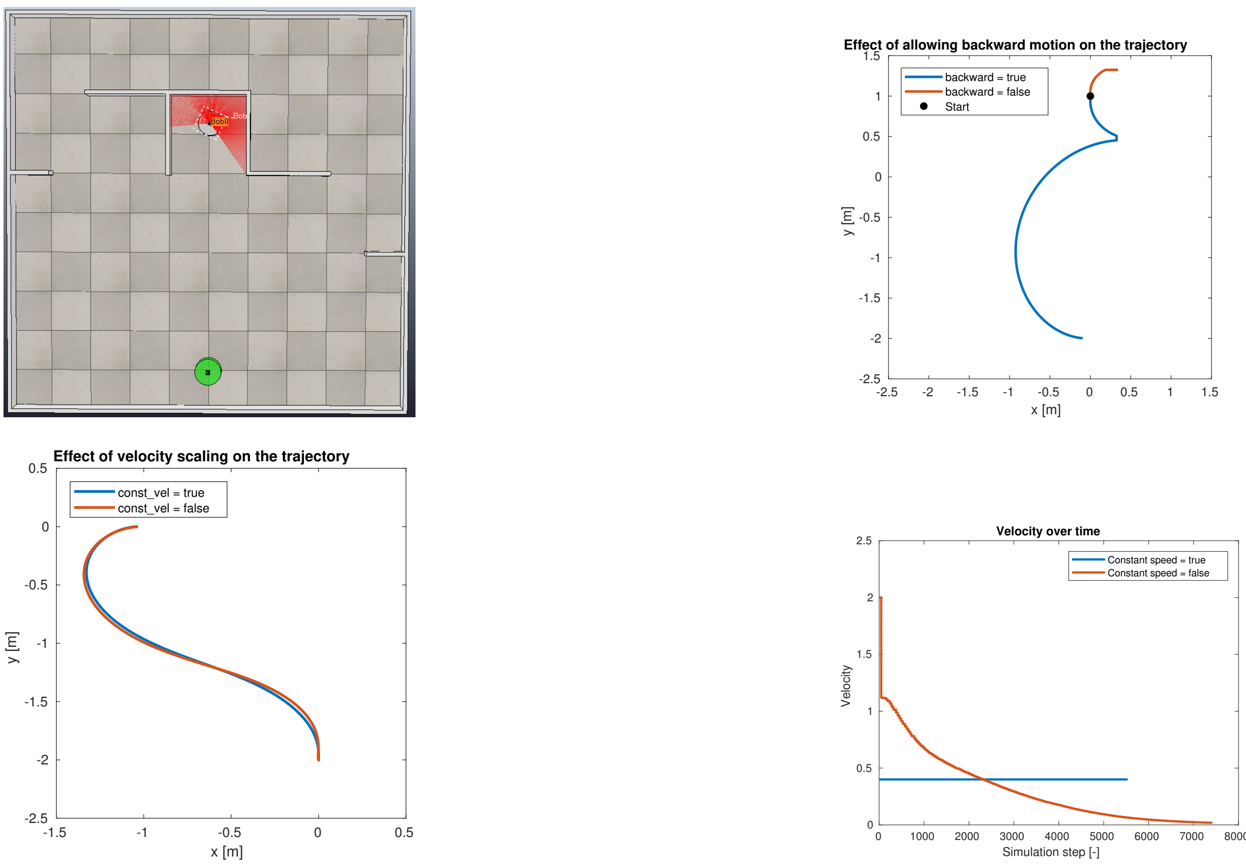

Figure 2: Top-left: topview of scene for testing the backward motion. Top-right: Resulting trajectory when backward motion is allowed. Bottom-left: Comparison of trajectory for constant velocity. Bottom-right: Velocity (v_u^*) for each simulation step