Simen Bootsma
27-03-2019

Small distances

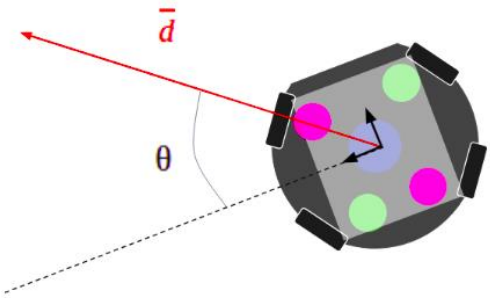
Goal
The goal of this experiment is to test how well the robot can drive small distances. This is an important feature for intercepting a ball or receiving a pass.

Method
In order to test this, we look at two quantities:

- $T(\underline{d})$, the time in which the robot reaches a certain position in seconds.
- $E(\underline{d})$, the steady-state error to the desired position after t_p seconds in meters.

Here, $\underline{d}$ is the vector from the initial position of the robot to the desired position. It is defined in polar coordinates with respect to the frame of the robot as follows:

$$\underline{d} = (d, \theta)$$



We have d as the length of the vector and θ the angle between the vector and the robot's positive v-axis.

A heat map for both functions $T(\underline{d})$ and $E(\underline{d})$ depending on the variables d and θwill give a nice overview of the strong points and the weak points of the robot. In this plot, you will be able to see where improvement is needed.

The measurements will be done by using GoToPos type Basic with a PID controller that has the values: P = 1.0, I = 0.0, D = 0.5. The error margin for the position control is set to 0.02 m. The robot is then instructed to go to a set of positions relative to itself. When it reaches that position, it takes a 2 second break and then moves on to the next position.

The movement of the robot is recorded using world state. It is not possible to start the time measurement when a command is sent, since it is very hard to get the ros topics */world_state* and */robotcommands* synced with each other. However, the tool PlotJuggler does manage to do this and using that, the time it takes for the robot to respond to a given command is estimated to be 10 ms. The total time $T(\underline{d})$ can now be measured by looking at the times when the measured velocity from the world state is greater than zero.

Therefore, we only need to record world state and the total time of recording to get all the necessary data.
Results

The test has been run both for driving left and driving right (relative to the robot). Driving left is shown in figure 1, where the range of $\theta$is from -20 to 20 degrees. Driving right is shown in figure 2, the range is 160 to -160 degrees there.
In the plots on the left, you can see $T(\underline{d})$ in the upper plot and $E(\underline{d})$ in the lower plot. Next to that, also the average velocity $\frac{|\underline{d}|}{T(\underline{d})}$ and the normalized position error $\frac{E(\underline{d})}{|\underline{d}|}$ have been shown.

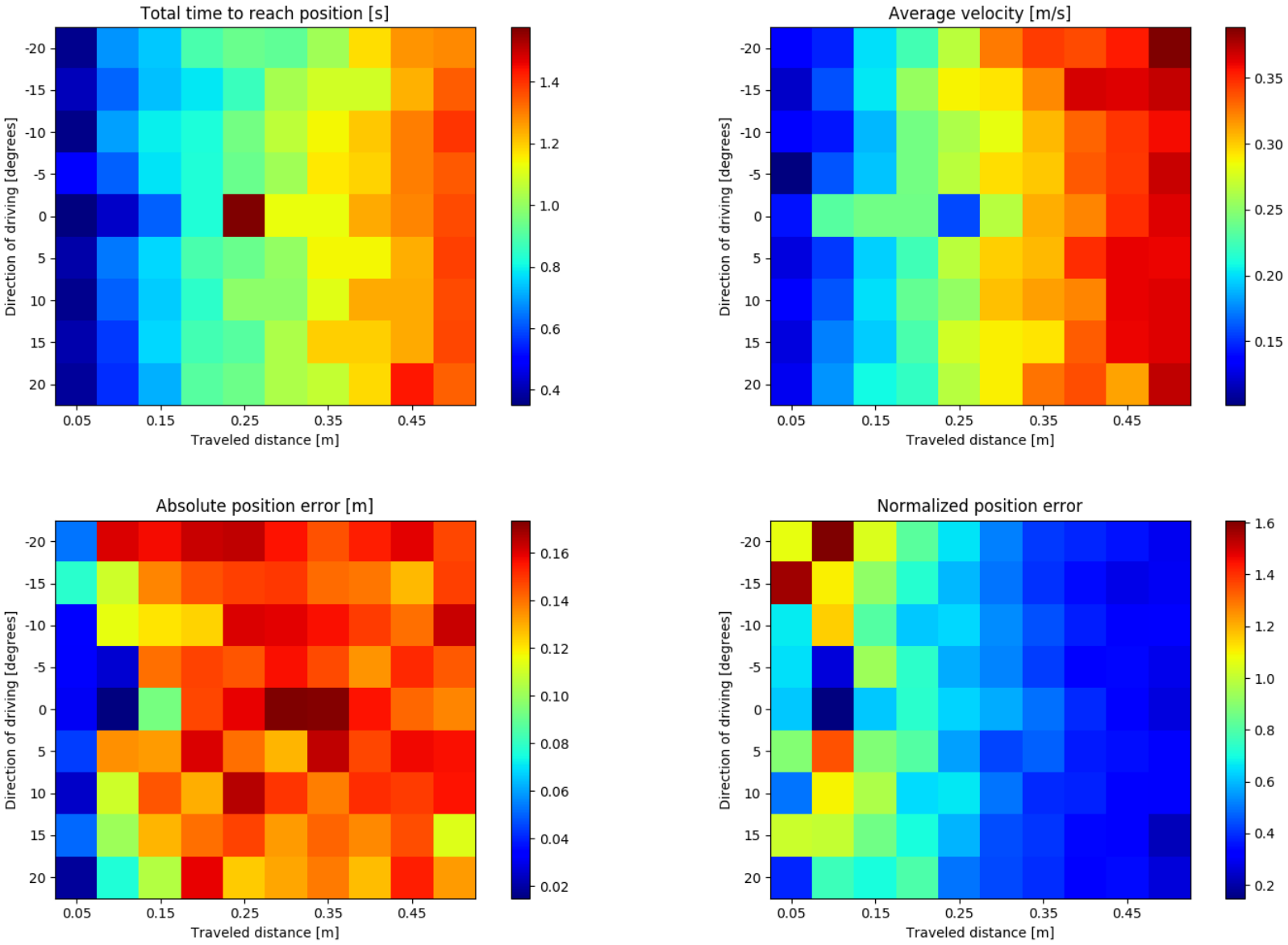


Figure 1: *Driving to the left of the robot.*

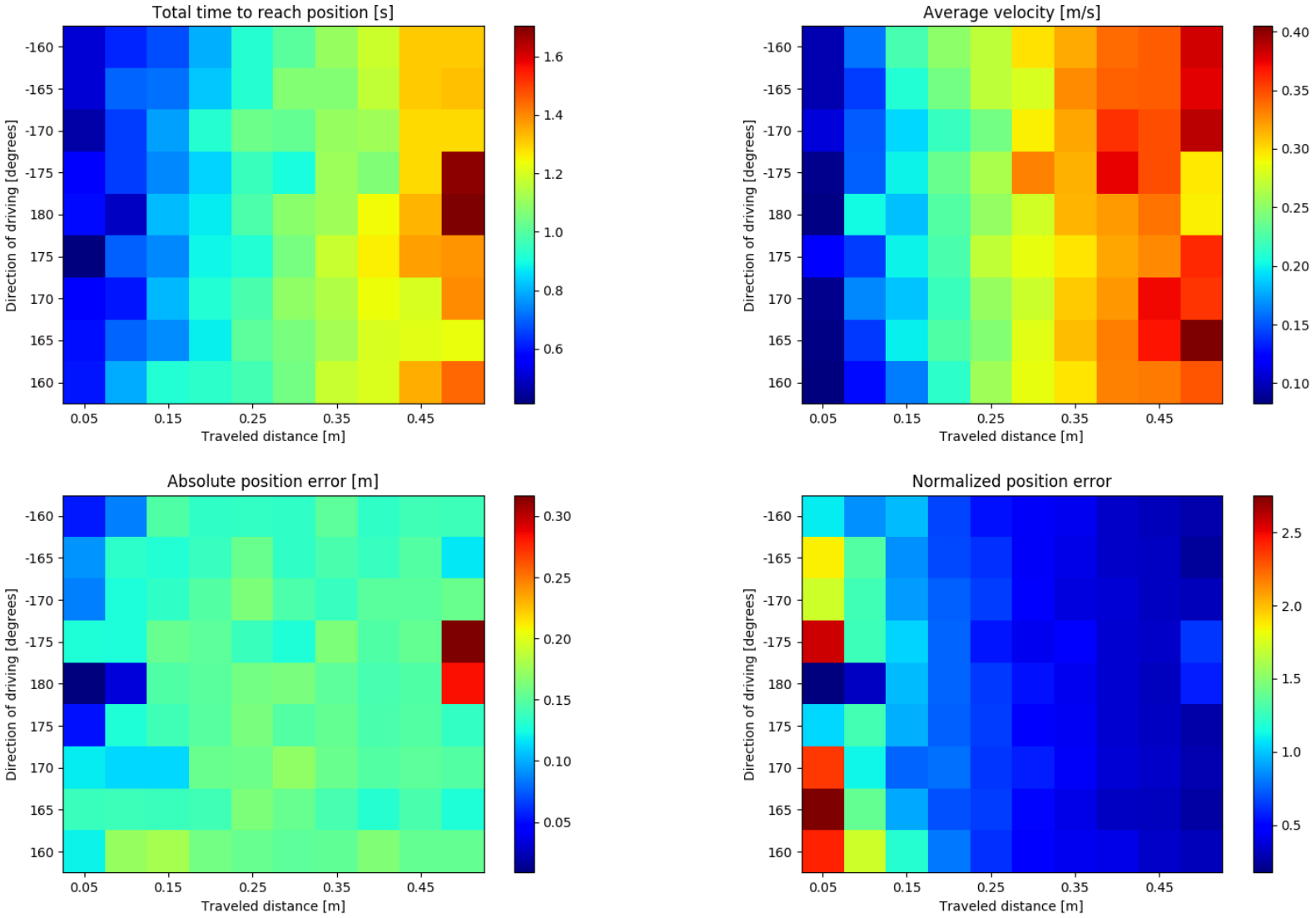


Figure 2: *Driving to the right of the robot.*

Two of the data points in the error in figure 2 are way higher than the rest, which makes the color different relative to figure 1. Other than that, both figures are essentially the same.

A clear difference cannot be seen between the different values of the direction of driving. The total time and the average velocity increase for increasing distances independent of the direction of driving. A clear relation between the direction of driving and the error can not be made as well.

The position error is quite large, about 15 cm on average. It is almost constant for all distances of 20 cm or greater. For the small distances, the error differs a lot. This does not seem to be dependent of the direction of driving, but rather some randomness in the behaviour.

Conclusion

At least when driving small distances (< 50 cm), it is not necessary to take into account the direction of driving. The position error is too large which should be fixed. The error behaviour is predictable for distances greater than 20 cm, but random for smaller distances. This should be looked at as well. Two different controllers might be a solution. Implementation of fuzzy logic might also be a possibility here.