Simen Bootsma
10-02-2019

Velocity control

Introduction

To achieve a given velocity command, the initial approach was to convert the global velocity to an angular velocity command for every wheel. Then make sure that the velocity command is satisfied by using PID control on the wheels. We thought that if the wheels meet their angular velocity commands, then the global velocity command is also met. In this document, we will show that this is not always the case.

Test case

First we gave a global velocity command of 1 m/s in the x-direction, while keeping the angle at 0 radians. This means that the robot is moving in its forward direction. This seemed to be done perfectly, so nothing special has happened.

Then we wanted to see if the same was true for moving in the y-direction, while keeping the angle at 0 radians (sideways). The absolute velocity in this case was 0.5 m/s, since the maximum velocity in this direction is less than in the forward direction due to the orientation of the wheels. The robot was instructed to move to its left for 1 second, then stop and move to its right for 1 second. This is where we could draw some conclusions from and its results are given below.

Calculations

To make sure that the wheels turn at the right velocities, the global velocity command has to be converted to an angular velocity command for every wheel. This is done by using two rotation matrices. The first one is used to rotate from a global to a local frame of reference. The second one is used to convert from local velocity to wheel speeds. These relations are given by the following equations:

$$v_{local} = R_1 v_{global} \tag{1}$$

$$\omega = R_2 v_{local} \tag{2}$$

$$R_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$R_2 = \begin{bmatrix} \frac{-1}{\sin(\frac{\pi}{3})} & \frac{-1}{\cos(\frac{\pi}{3})} \\ \frac{-1}{\sin(\frac{\pi}{3})} & \frac{1}{\cos(\frac{\pi}{3})} \\ \frac{1}{\sin(\frac{\pi}{3})} & \frac{1}{\cos(\frac{\pi}{3})} \\ \frac{1}{\sin(\frac{\pi}{3})} & \frac{-1}{\cos(\frac{\pi}{3})} \end{bmatrix} \frac{1}{r}$$

, where r is the radius of the wheels in meters.

Results

We measured the wheel speed commands that have been sent to the wheels (reference) together with the actual wheel speeds. These are shown together in figure 1. It is clear that the wheels reach their references pretty well, but there are quite some fluctuations in the wheel speeds. This is probably due to the friction of the field in different directions.

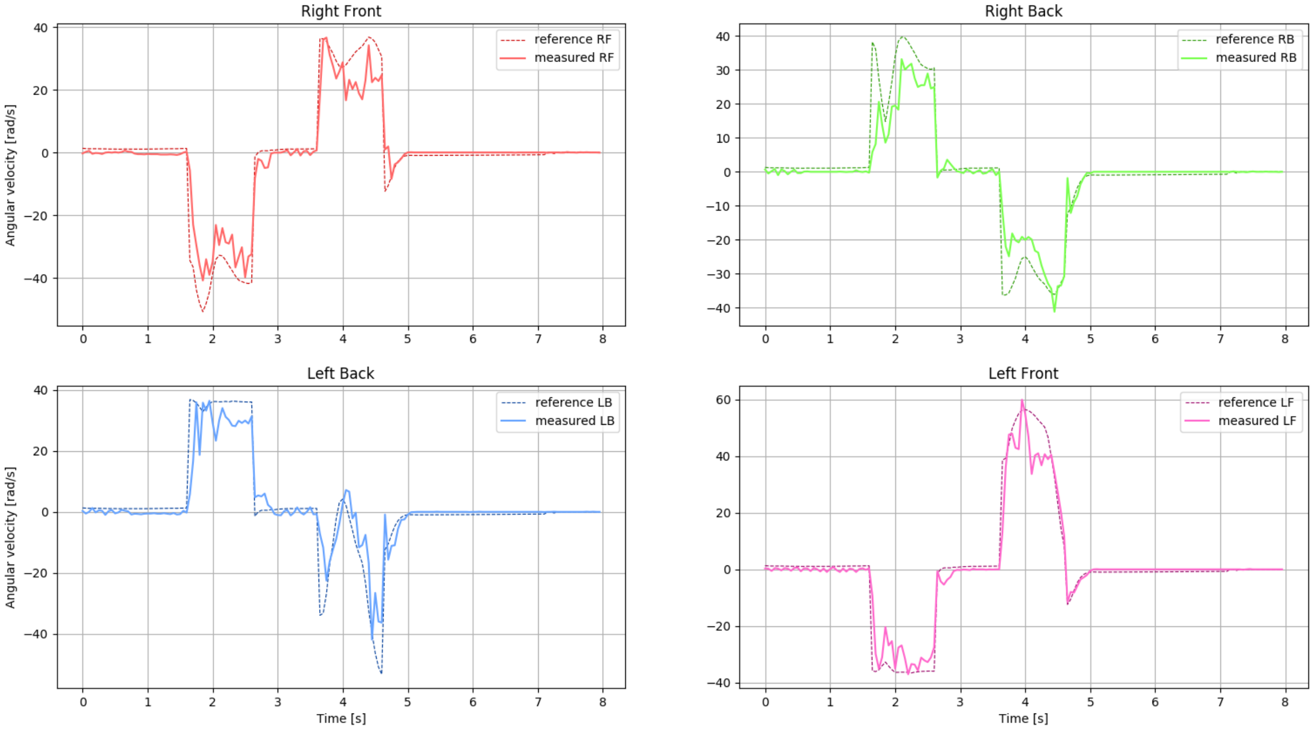

Figure 1: *Wheel speeds as instructed by the control and measured using encoders.*

Furthermore, we measured the PWM that is sent to the wheels. This PWM consists of the wheel speed reference and the output of the PID. This is shown for the different wheels in figure 2. It can be seen that the maximum PWM is only reached for the initial step input, so the system should be functioning fine. Also, the fluctuations from the wheel speeds can be seen in the PWM graph as well.

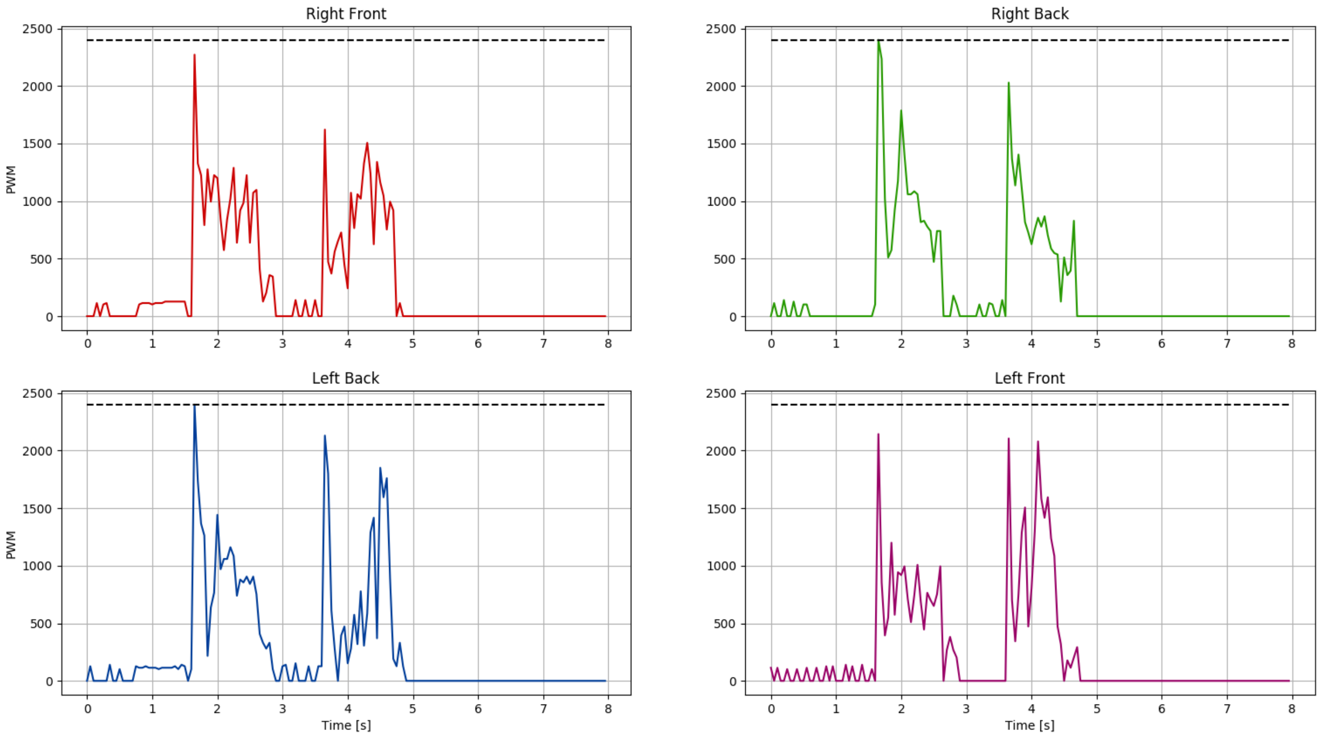

Figure 2: *PWM that is sent to the different wheels, maximum is depicted with the black line.*

Using equations (1) and (2), the global velocities of the robot can be calculated from measured wheel data. This is shown in figure 3. The robot is given a velocity command of 0.5 m/s in the y-direction. This can be seen in the graph and the robot seems to reach this velocity very well. However, in the x-velocity graph can be seen that the robot reaches velocities up to 0.3 m/s in this direction while it did not receive the command to do this.

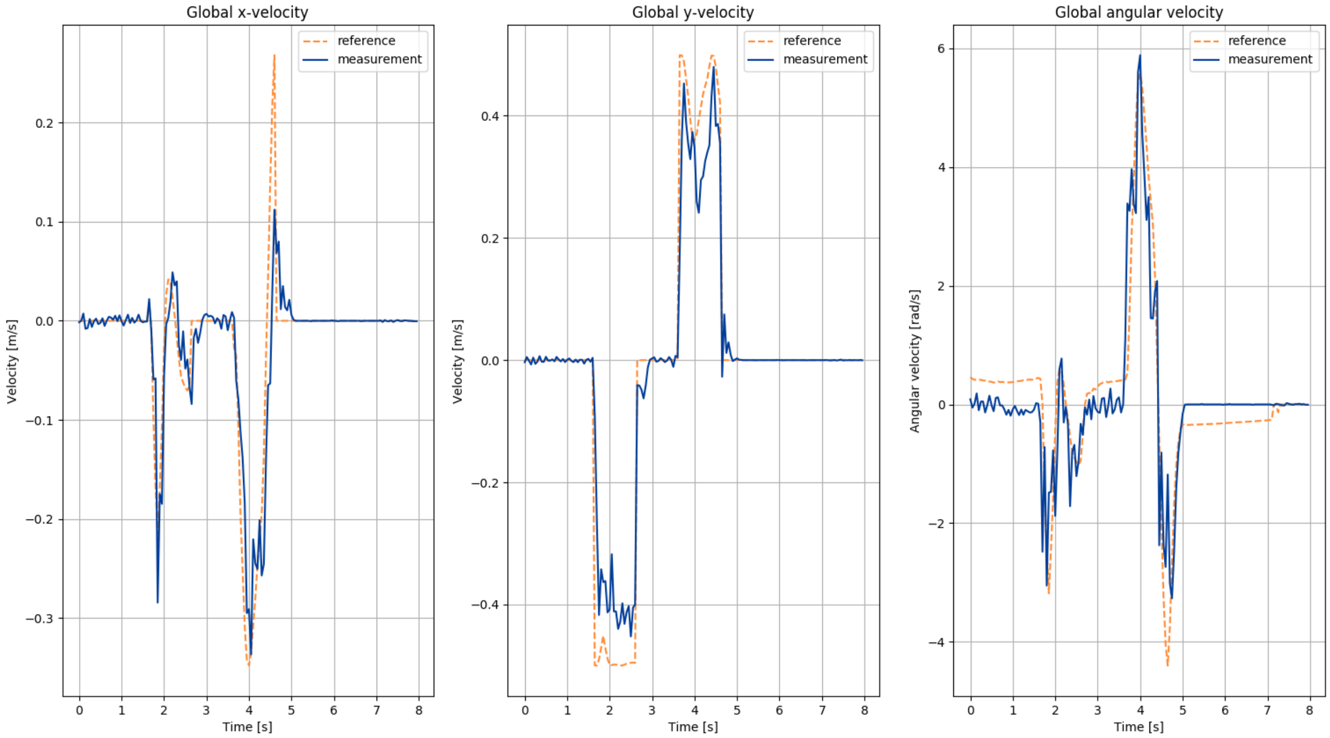

Figure 3: *Global velocities of the robot as calculated from the wheels frame.*

Conclusion

Although the robot responds well to velocity commands in the forward direction, the results show that the robot moves a lot in the forward direction while it is supposed to move in the sidewards direction. This means that next to wheel control, also a form of velocity control is necessary to make sure that the robot only moves in the direction it is supposed to move.This is an expected, but unfortunate result, since velocity control requires a good state estimation. Therefore, data from the wheel encoders should be combined with data from the accelerometer and optionally vision data.