Kinematics means to compute the speed of each wheel out of the bots movement and vice versa.

I wont bother you with the details oft wo weekend’s work. To make a long story short, the final formula is:

which gives you the speed of every omniwheel out of the speed in x/y direction and the angular speed around the z-axis.

The system-specific variables used are

|  |  |  |
| --- | --- | --- |
|  | angle between horizontal plane and the omniwheels directions [rad] | 45° = |
|  | radius of the omniwheel in [mm] | 35mm |
|  | radius of the ball in [mm] | 180mm |

The kinematic parameters are

|  |  |
| --- | --- |
|  | angular velocity of the i.th wheel = |
|  | speed in x-direction in |
|  | speed in y-direction in |

In the formula above, R is the rotation matrix of the current tilt of the bot, it is a regular rotation matrix

where

|  |  |
| --- | --- |
|  | tilt angle of the bot in y direction in [rad] |
|  | tilt angle of the bot in x direction in [rad] |

For forward kinematics we need the formula reversed, which is

This gives you the speed in x and y direction out oft he speed of all omniwheels

Implementation

During setup of the bot, we can precompute the so-called construction matrix CM.

During runtime a continously running loop, we have the tilt angles coming from the IMU, and compute the tilt correction matrix R. We are lucky that al that is finally multiplied with the sparse matrix

which means that we do not have to compute a full matrix multiplication of CM\*R with 81 floating point multiplications, but can omit the computation where the matrix above has a 0 requiring 10 multiplications only.