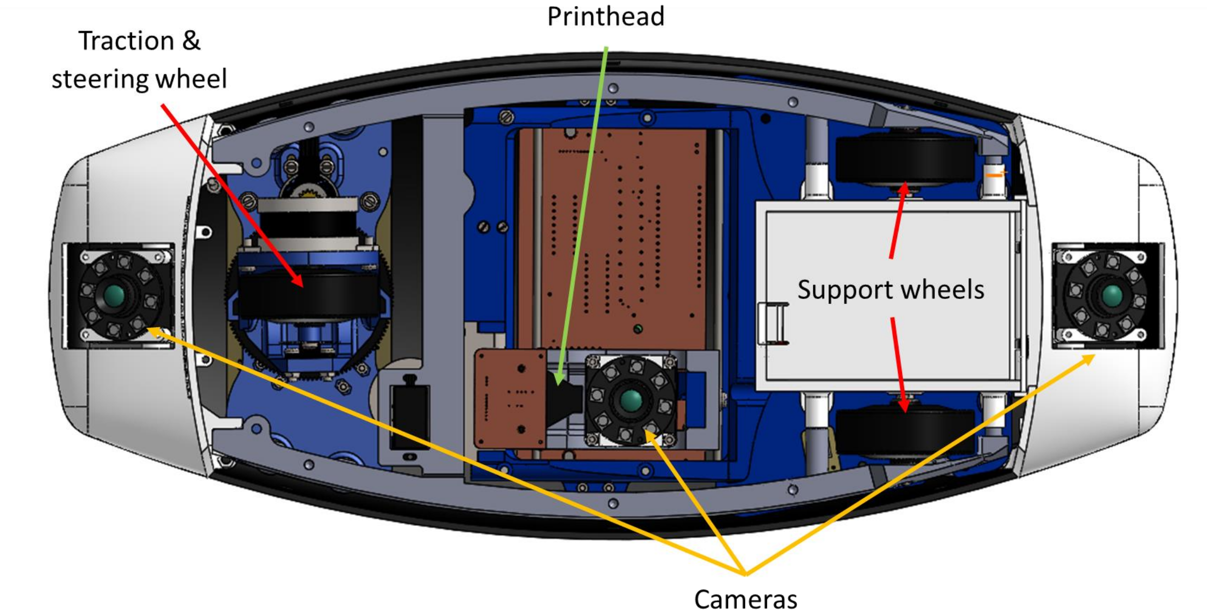
Scriba

A printing Robot by Robin Baran

This document aim to describe all technical aspects of the Scriba robot.

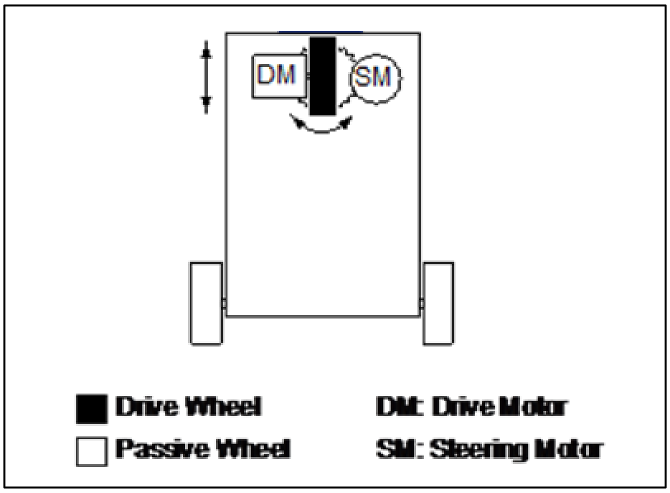
The Scriba robot is a printer robot able to navigate on a large and flate medium and able to print on its wake, in black and white. It uses multiple cameras to correct its trajectory and printing alignment through a SLAM algorithm.

1 General characteristics



1.1 Drive configuration

The robot has a steer drive configuration with one powered and steering wheel at the front and 2 wheel for support at the rear. This configuration was chosen because of the high driving accuracy it provides when following a navigation path.



The driving wheel is powered by a stepper motor (14HR08-0654S) with a step angle resolution of 0.9 deg. In addition an encoder counts the rotation angle with a resolution of 1.8deg.

The driving wheel is steered by a stepper motor (14HR05-0504S) with a step angle resolution of 0.9 deg. In addition an encoder counts the rotation angle with a resolution of 1.8deg. The system is designed to rotate the driving wheel +/-90°.

1.2 Steering system

TODO

1.3 Printing system

1.3.1 Printhead

The printhead used by the robot is an inkjet printhead HP 51604A. It has been chosen for its ease of use and low price. Hacked inkjet printhead with easy controls are difficult to find. The HP 51604A (and the HP55*****) have a good document available online.

1.3.2 Alignment & Correction

1.3.3 Printing slider

2 Kinematics

2.1 General equations

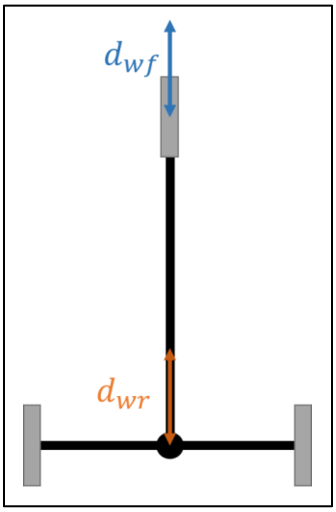
The kinematics equations of the robot are needed to implement move command and odometry. ROS move command is sent as a velocity (kinematic center speed) and twist (angular speed) command.

There are two main configurations to consider:

- Steering wheel with no angle
- Steering wheel with angle

In both cases we want to express the angle of rotation and the kinematic center displacement relative to the front wheel angle and front wheel displacement.

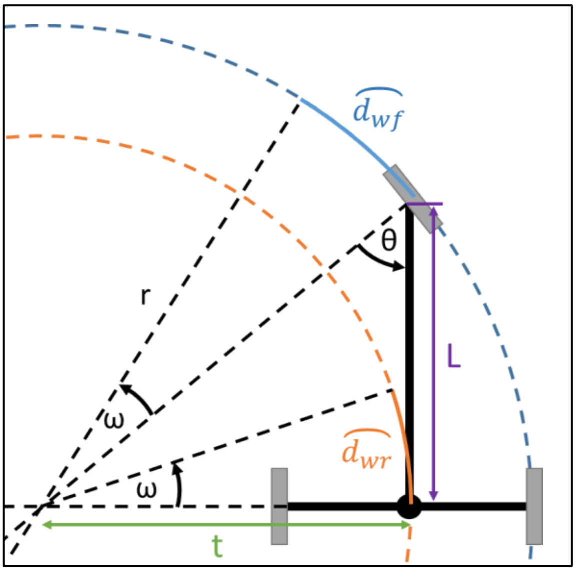
2.1.1 Steering wheel with no angle



It's easy to get the angle of rotation and the kinematic center displacement relative to the front wheel angle and front wheel displacement:

- Angle of rotation: $\omega = 0$
- Kinematic center displacement: $d_{wr} = d_{wf}$

2.1.2 Steering wheel with angle



$$\tan(\theta) = \frac{t}{L} \quad \Leftrightarrow \quad t = L * \tan(\theta)$$

$$\cos(\theta) = \frac{L}{r} \quad \Leftrightarrow \quad r = \frac{L}{\cos(\theta)}$$

$$\widehat{d_{wf}} = \omega * r = \frac{\omega * L}{\cos(\theta)} \quad \Leftrightarrow \quad \omega = \widehat{d_{wf}} * \frac{\cos(\theta)}{L}$$

$$\widehat{d_{wr}} = \omega * t = \omega * L * \tan(\theta) = \widehat{d_{wf}} * \frac{\cos(\theta)}{L} * L * \tan(\theta) = \widehat{d_{wf}} * \cos(\theta) * \tan(\theta)$$

From those equations we obtain the angle of rotation and the kinematic center displacement relative to the front wheel angle and front wheel displacement:

- Angle of rotation: $\omega = \widehat{d_{wf}} * \frac{\cos(\theta)}{L}$
- Kinematic center displacement: $\widehat{d_{wr}} = \widehat{d_{wf}} * \cos(\theta) * \tan(\theta)$

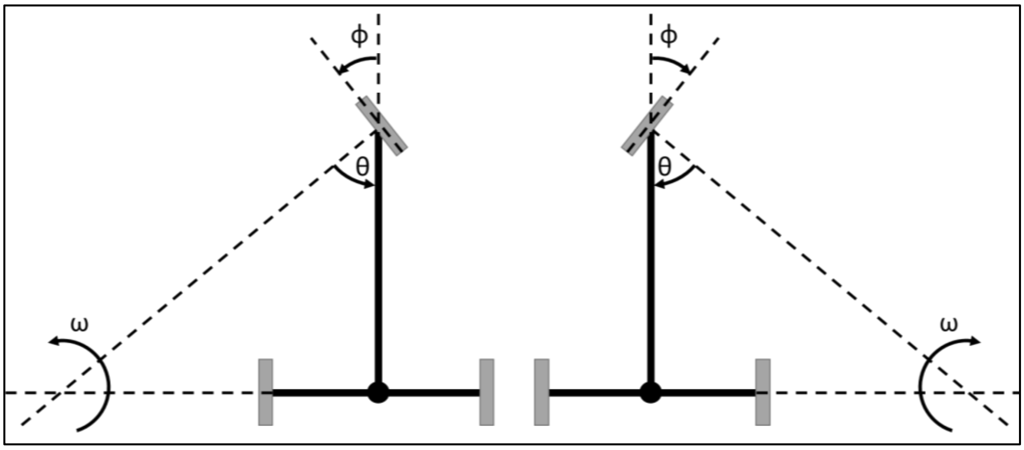
2.2 Velocity command

ROS command outputs a kinematic center speed and an angular speed. On the other end the motor controls need a front wheel angle and a front wheel speed.

Therefore, command for the robot has to be expressed as a front wheel angle and front wheel speed relative to the angular speed and the kinematic center speed.

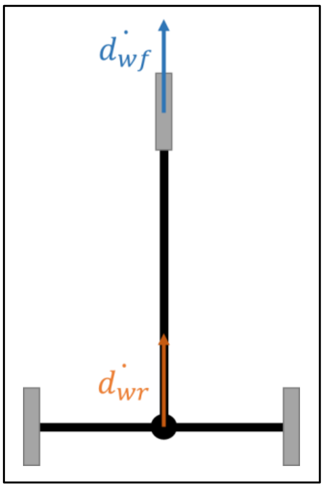
The front wheel angle sent as a motor command is ϕ and is expressed as:

$$\varphi = \frac{\pi}{2} - \theta \quad \Leftrightarrow \quad \theta = \frac{\pi}{2} - \varphi$$



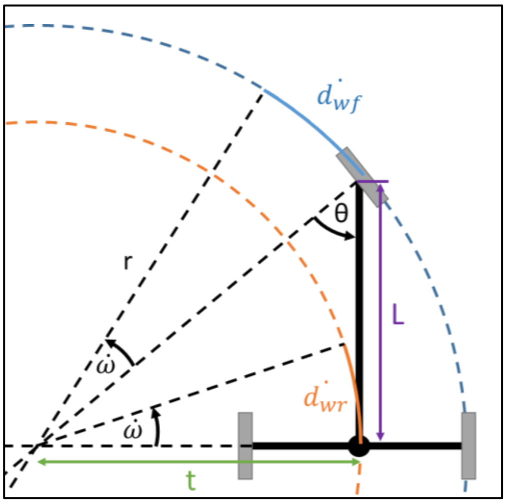
2.2.1 No angular speed command

In the special case of a command with no angular speed.



- Front wheel rotation: $\varphi = 0$
- Front wheel speed: $\dot{d_{wf}} = \dot{d_{wr}}$

2.2.2 With angular speed command



Using the previous equations (2.1.2 Steering wheel with angle - page 5) the following equations are derived:

$$\widehat{d_{wr}} = \omega L \tan(\theta)$$
$$\Leftrightarrow \dot{d_{wr}} = \dot{\omega} L \tan(\theta) + \dot{\omega} L \dot{\theta}(1 + tan^2(\theta))$$

And

$$\widehat{d_{wr}} = \widehat{d_{wf}} \cos(\theta) \tan(\theta)$$
$$\dot{d_{wr}} = \dot{d_{wf}} \cos(\theta) \tan(\theta) + \widehat{d_{wf}}\, \dot{\theta}\left(\cos(\theta)\left(1 + tan^2(\theta)\right) - \sin(\theta)\tan(\theta)\right)$$
$$\dot{d_{wr}} = \dot{d_{wf}} \sin(\theta) + \widehat{d_{wf}}\, \dot{\theta}(\cos(\theta) + \sin(\theta)\tan(\theta) - \sin(\theta)\tan(\theta))$$
$$\dot{d_{wr}} = \dot{d_{wf}} \sin(\theta) + \widehat{d_{wf}}\, \dot{\theta}\cos(\theta)$$

Θ is considered constant for several reasons. Controlling both the front wheel angle and the front wheel angular speed is unpractical. Indeed the base controller would need the exact front wheel angle and the communication time with the Arduino sending that data is too long. Moreover, the front wheel angular speed will be very small in practice.

If θ constant then $\dot{\theta} = 0$, the previous equations can be rewritten:

$$\dot{d_{wr}} = \dot{\omega} L \tan(\theta) = \dot{\omega} L tan\left(\frac{\pi}{2} - \varphi\right) = \frac{\dot{\omega} L}{\tan(\varphi)} \Leftrightarrow \tan(\varphi) = \frac{\dot{\omega} L}{\dot{d_{wr}}} \Leftrightarrow \varphi = atan\left(\frac{\dot{\omega} L}{\dot{d_{wr}}}\right)$$

$$\dot{d_{wr}} = \dot{d_{wf}} \cos(\theta)\tan(\theta) = \dot{d_{wf}} \cos\left(\frac{\pi}{2} - \varphi\right) \tan\left(\frac{\pi}{2} - \varphi\right) = \dot{d_{wf}} \sin(\varphi) \frac{\cos(\varphi)}{\sin(\varphi)}$$
$$= \dot{d_{wf}} \cos(\varphi) \Leftrightarrow \dot{d_{wf}} = \frac{\dot{d_{wr}}}{\cos(\varphi)}$$

- Front wheel rotation: $\varphi = atan\left(\frac{\dot{\omega} L}{\dot{d_{wr}}}\right)$
- Front wheel speed: $\dot{d_{wf}} = \frac{\dot{d_{wr}}}{\cos(\varphi)}$

2.2.3 Only angular speed command

Another special case is needed is only angular speed is required. Indeed, for an angle of $\varphi = \frac{\pi}{2} \longrightarrow \cos(\varphi) = 0$ and $\dot{d_{wf}} = \frac{\dot{d_{wr}}}{\cos(\varphi)}$ would go to infinite.

Using the previous equations (2.1.2 Steering wheel with angle - page 5) the following equations are derived:

$$\omega = \widehat{d_{wf}} * \frac{\cos(\theta)}{L}$$

$$\dot{\omega} = \dot{d_{wf}} * \frac{\cos(\theta)}{L} + \frac{\widehat{d_{wf}}}{L} * \left(-\dot{\theta}\sin(\theta)\right)$$

If θ constant then $\dot{\theta} = 0$, the previous equations can be rewritten:

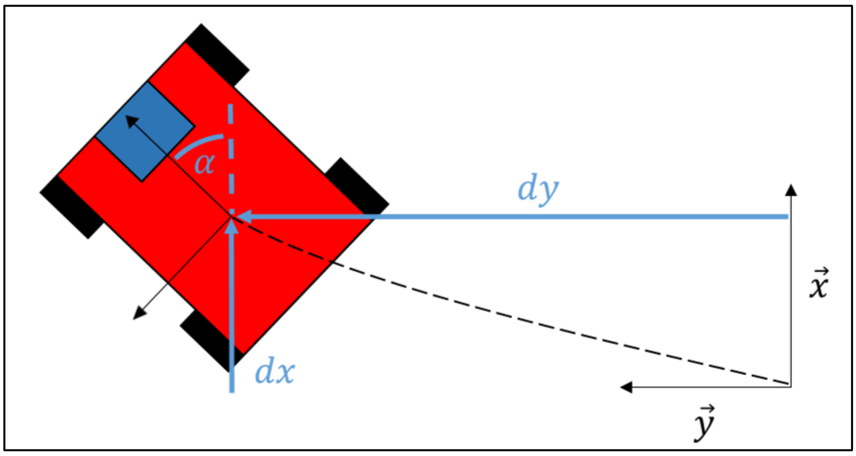
$$\dot{\omega} = \dot{d_{wf}} * \frac{\cos(\theta)}{L} = \dot{d_{wf}} * \frac{\sin\left(\frac{\pi}{2} - \varphi\right)}{L} = \dot{d_{wf}} * \frac{\sin(\varphi)}{L}$$

$$\Leftrightarrow \quad \dot{d_{wf}} = \frac{\dot{\omega} L}{\sin(\varphi)}$$

- Front wheel rotation: $\varphi = \frac{\pi}{2} \; or - \frac{\pi}{2}$
- Front wheel speed: $\dot{d_{wf}} = \frac{\dot{\omega} L}{\sin(\varphi)}$

2.3 Odometry

In ROS, a 2D odometry is expressed as a displacement on the x-axis, a displacement on the y-axis and a rotation around the z-axis, all expressed in a global odometry frame.



The displacement will be first expressed in the robot base frame:

- ω
- Δx
- Δy

And then added to the odometry frame using the following equations:

- $\alpha = \alpha + \omega$
- $dx = dx + \Delta x * \cos(\alpha) - \Delta y * \sin(\alpha)$
- $dy = dy + \Delta x * \sin(\alpha) + \Delta y * \cos(\alpha)$

2.3.1 No angular speed

Without angular speed the odometry is very straight-forward:

- $\omega = 0$
- $\Delta x = d_{wf}$
- $\Delta y = 0$

2.3.2 With angular speed

Using the previous equations (2.1.2 Steering wheel with angle - page 5) the following equations are derived:

$$\omega = \widehat{d_{wf}} * \frac{\cos(\theta)}{L} = \widehat{d_{wf}} * \frac{\sin(\varphi)}{L}$$

And

$$\widehat{d_{wr}} = \widehat{d_{wf}} \cos(\theta) \tan(\theta)$$

$$\dot{d_{wr}} = \dot{d_{wf}} \cos(\theta) \tan(\theta) + \widehat{d_{wf}}\, \dot{\theta}\left(\cos(\theta)\left(1 + tan^2(\theta)\right) - \sin(\theta) \tan(\theta)\right)$$

$$\dot{d_{wr}} = \dot{d_{wf}} \sin(\theta) + \widehat{d_{wf}}\, \dot{\theta}(\cos(\theta) + \sin(\theta) \tan(\theta) - \sin(\theta) \tan(\theta))$$

$$\dot{d_{wr}} = \dot{d_{wf}} \sin(\theta) + \widehat{d_{wf}}\, \dot{\theta} \cos(\theta)$$

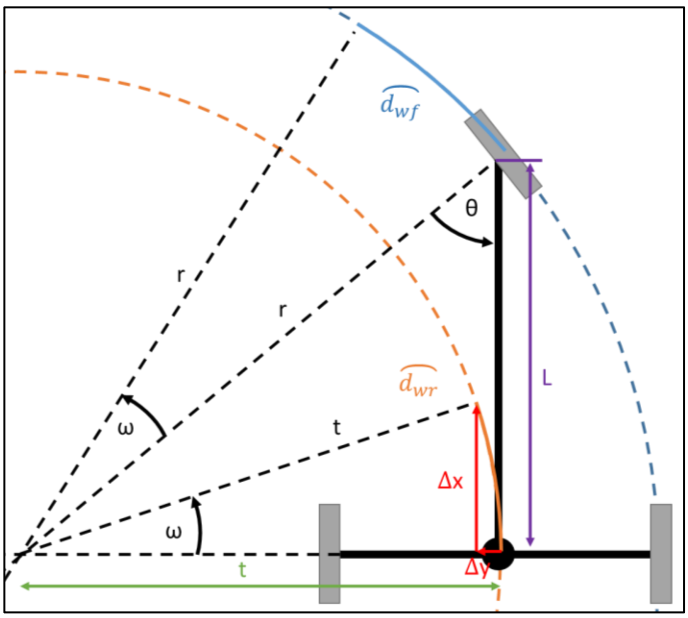
As before, Θ is considered constant. The odometry is computed from the front wheel displacement ($\widehat{d_{wf}}$) in a defined time period (Δt), the front wheel angle (φ) and displacement in the same defined time period. All that data is received from the motor controller board.

We can express the displacement and angle according to the data:

$$\dot{d_{wr}} = \dot{d_{wf}} \sin(\theta) = \dot{d_{wf}} \cos(\varphi)$$

$$d_{wr} = \dot{d_{wr}} * \Delta t$$

$$\omega = \widehat{d_{wf}} * \frac{\sin(\varphi)}{L}$$



The odometry can then be deducted from the previous equations:

- $\omega = \widehat{d_{wf}} * \frac{\sin(\varphi)}{L}$

- $\Delta x = L\, tan(\theta) \sin(\omega) = L\, cotan(\varphi) \sin(\omega) = L\, \frac{\cos(\varphi)}{\sin(\varphi)} \sin(\omega)$
- $\Delta y = t - t\cos(\omega) = t * (1 - \cos(\omega)) = L\, \frac{\cos(\varphi)}{\sin(\varphi)} (1 - \cos(\omega))$