SCUTTLE Robot Kinematics Guide

revised 2025.06
Author: David Malawey

Contents

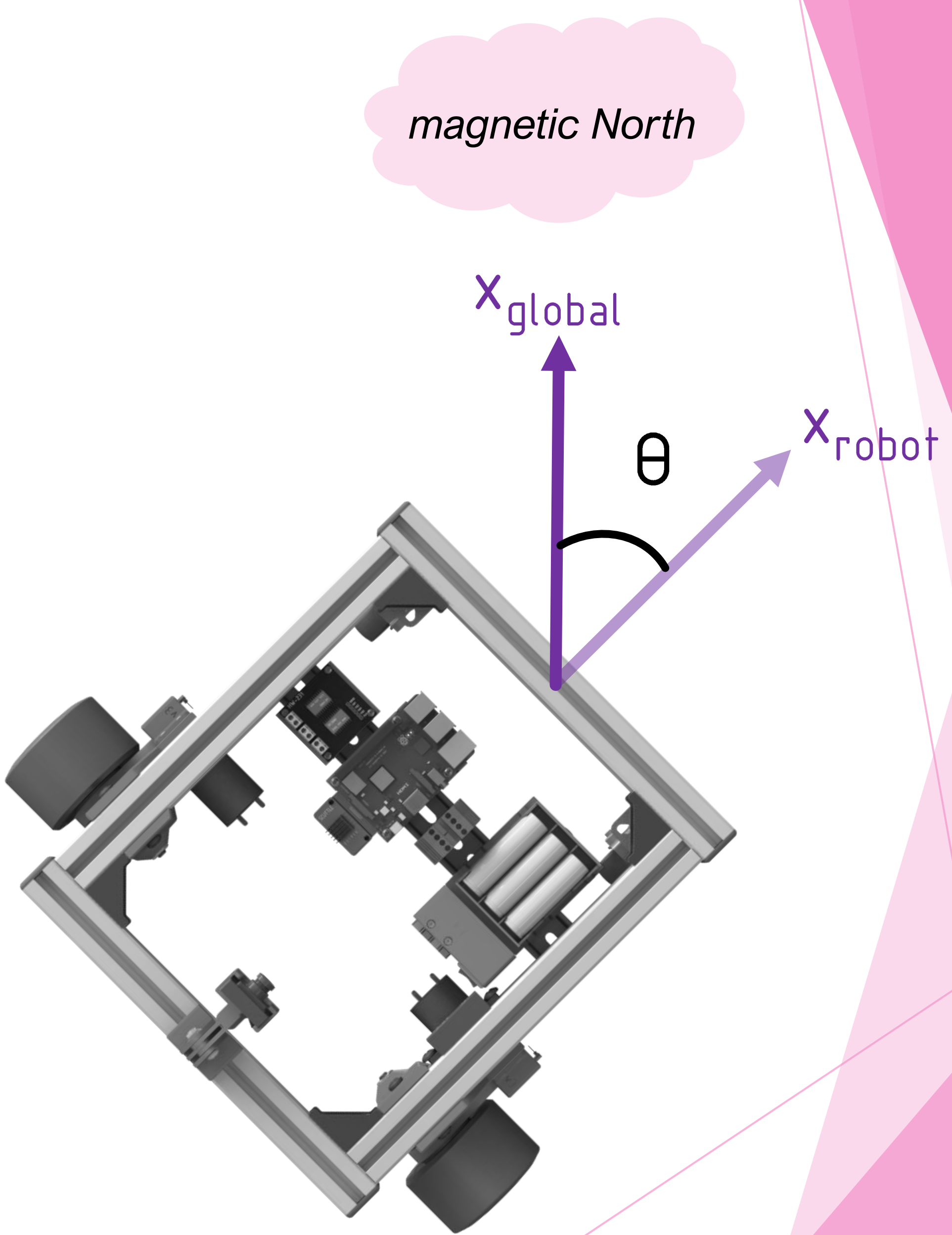
This Guide Covers:

- Robot geometry, r, and L
- **Key variables**: phi, x, y, and theta
- **Kinematic equation**: convert wheel speeds to chassis speeds
- **Inverse Kinematic equation**: convert chassis speeds to wheel speeds
- **Time-derivatives** of the wheel and chassis displacements
- **Rotation matrix** to convert body-fixed coordinates to global coordinates

SCUTTLE Coordinates

Our Axes & Coordinates:

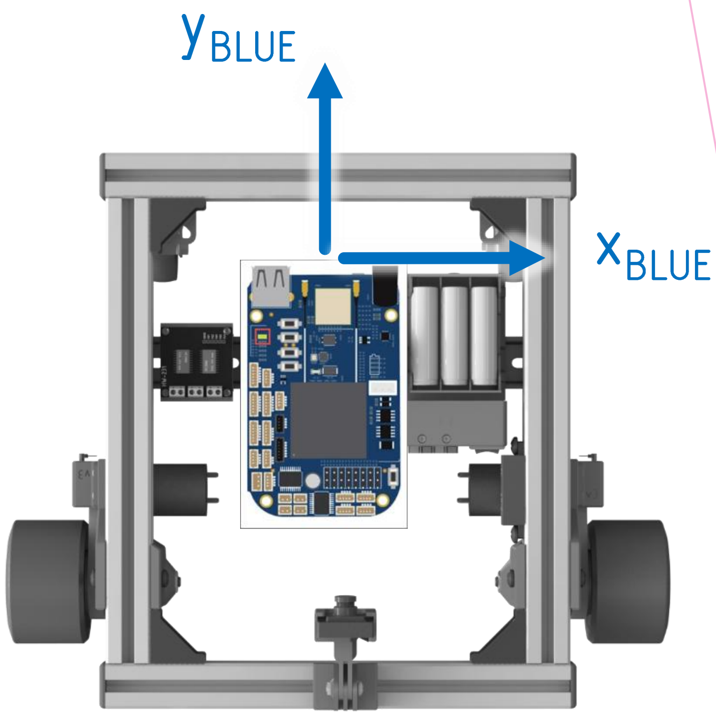
- The x-global is often decided to be aligned with magnetic North.
- The x-robot is aligned with the chassis, forward
- In this diagram, x-global is the axis and x-scuttle is bot axis
- Theta describes the orientation of the bot, in global coordinates



IMU Sensor

If your bot has an IMU (inertial measurement unit) then:

- The IMU reports values in the orientation of itself.
- You must translate or rotate the IMU reading if the sensor is mounted differently than forward.
- For beaglebone blue, the IMU has a y-axis pointing along the USB port.



The beaglebone blue includes an IMU with the y-axis as shown.

Non-Holonomic System

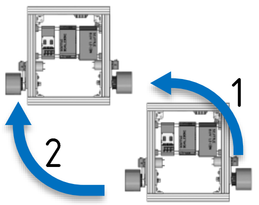
A holonomic robot has the same number (or more) of controllable degrees of freedom as the number of degrees of freedom.

SCUTTLE DOF: (x, y, theta)
CONTROLLABLE DOF: (left motor, right motor)
Mecanum Robot DOF: (x,y,theta)

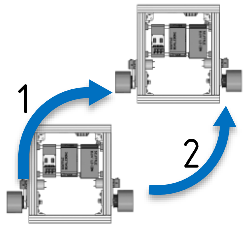
In a non-holonomic system, the final position of the robot depends on the path taken to achieve the movement.

Example:

Move right wheel, then left

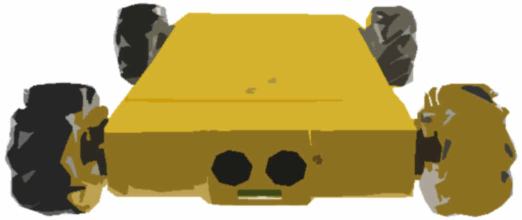


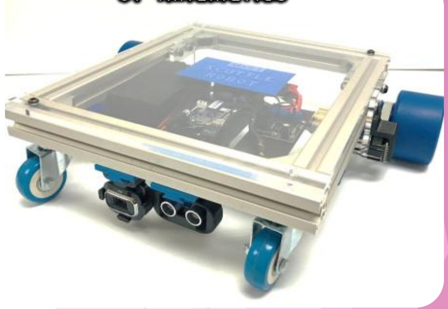
Move left wheel, then Right



Suitable for beginners

Requires knowledge of kinematics





Holonomic:

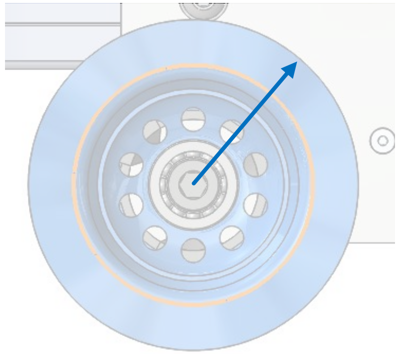
Easier navigation
More parts
Less robust

Non-Holonomic:

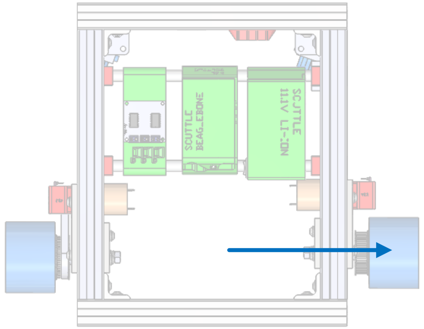
Harder navigation
Fewer parts
More robust

Chassis Geometry

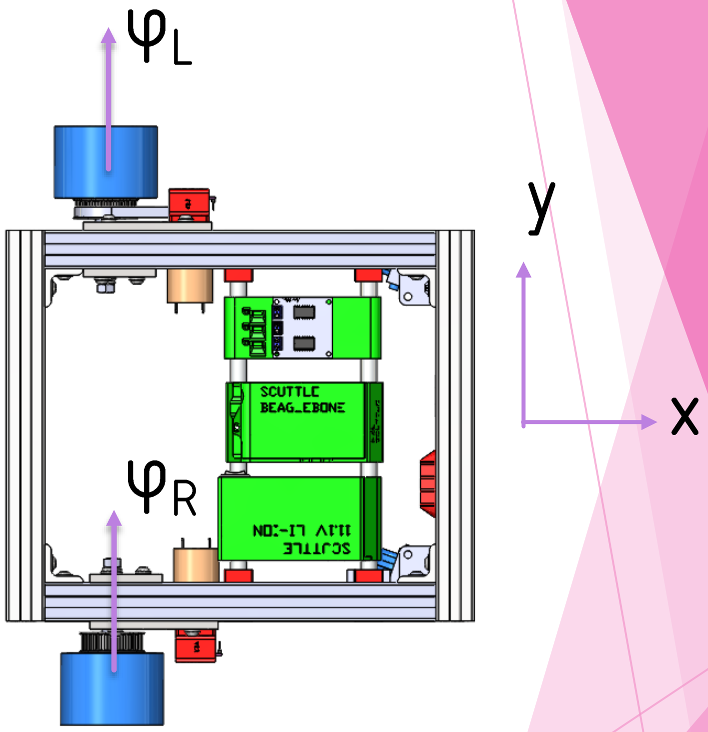
- The **Chassis Geometry** determines the equations for kinematics.
- The **radius**, r, is the radius of the driven wheel
- The **half-wheelbase**[(1)], L, is the space from wheel center to center divided in two



R = 41 mm



L = 201 mm [(1)]



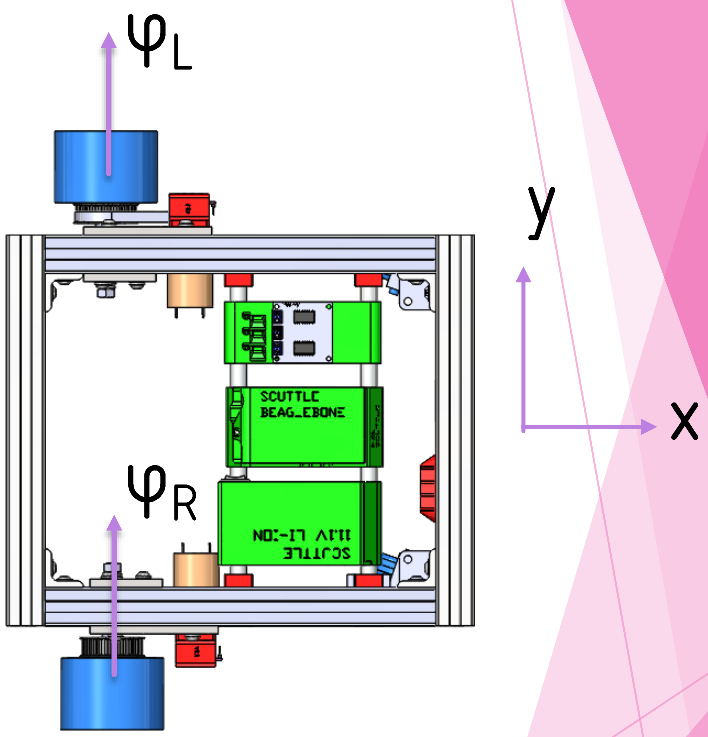
Parameters

- Phi dot is the derivative of phi with respect to time.

$\dot{\varphi}_L$ = pdl, as in phi_dot_l

$\dot{\varphi}_R$ = pdr, as in phi_dot_r

```
phiDots = np.array( [ pdl, pdr ] ) # python syntax
```

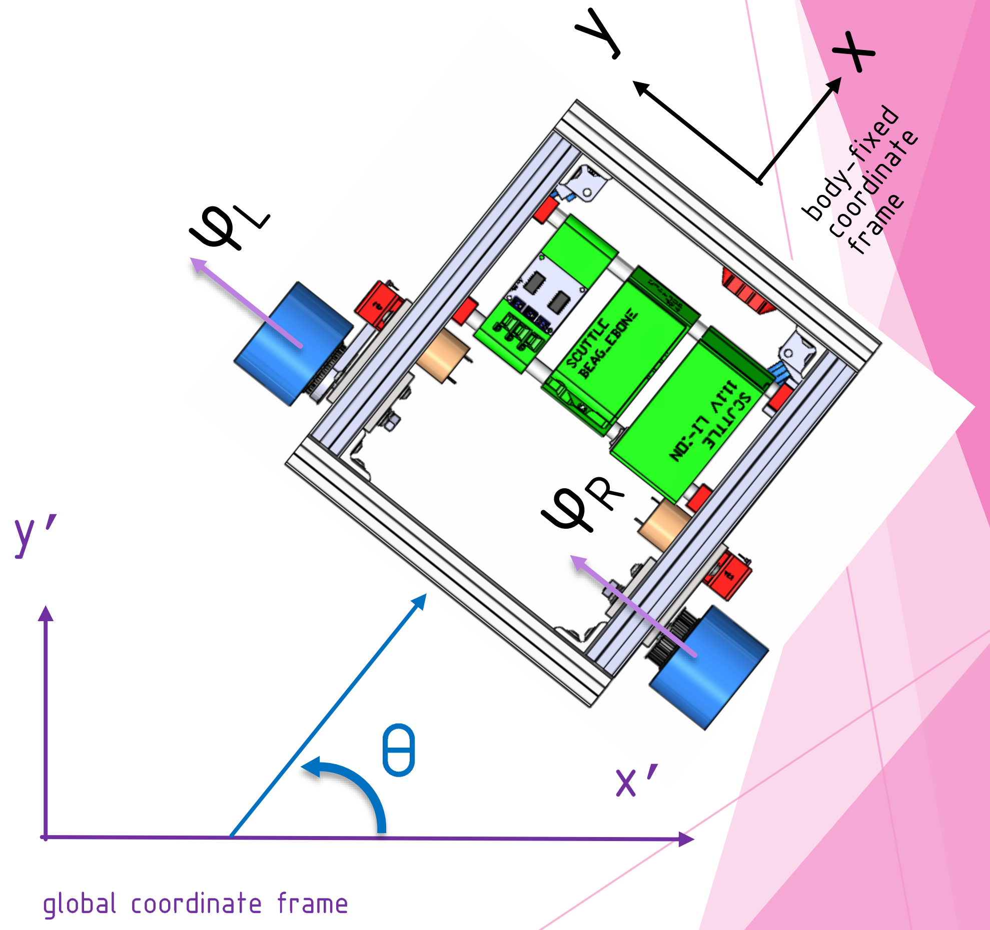


Rotation Matrix

- **Theta** describes the difference between the body-fixed frame and the global frame.

- The **rotation matrix** converts body-fixed coordinates to the global coordinates

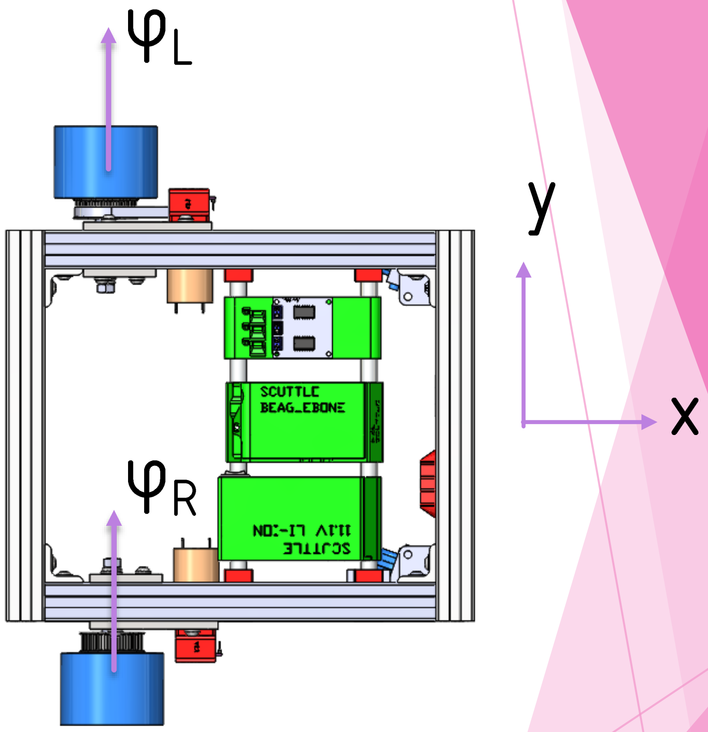
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_{bf} \\ y_{bf} \end{bmatrix}$$



Kinematic Equation

- Phi is the angle of the wheel.
  - It is used to define incremental changes in wheel position and to calculate wheel speeds
- The x,y coordinate system has x pointing forward on the bot.
  - Positive movement of both phi's result in positive movement of the robot along the x-direction
- The Kinematic Equation generates chassis motion information.
  - input the wheel speeds and output the (translational and rotational) chassis speeds

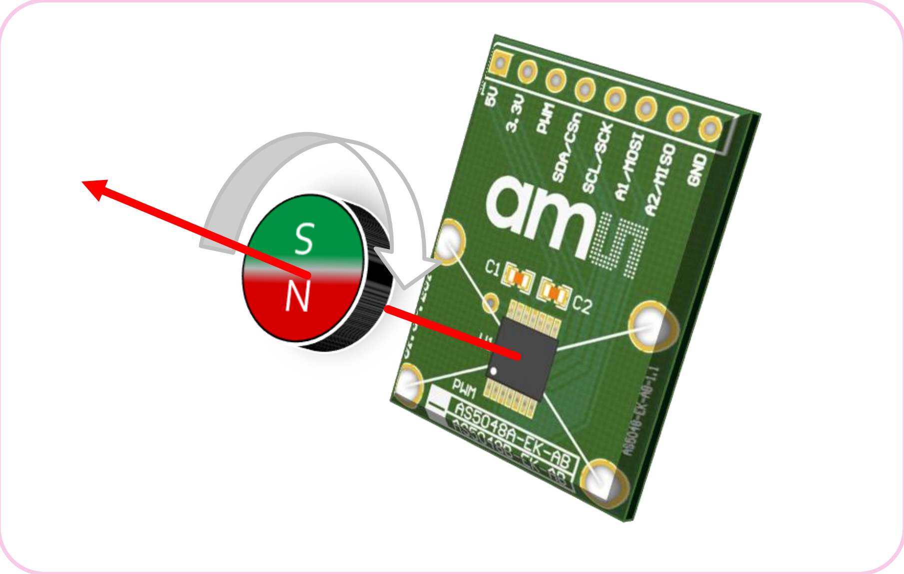
$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ -R/2L & R/2L \end{bmatrix} \begin{bmatrix} \dot{\varphi}_L \\ \dot{\varphi}_R \end{bmatrix}$$



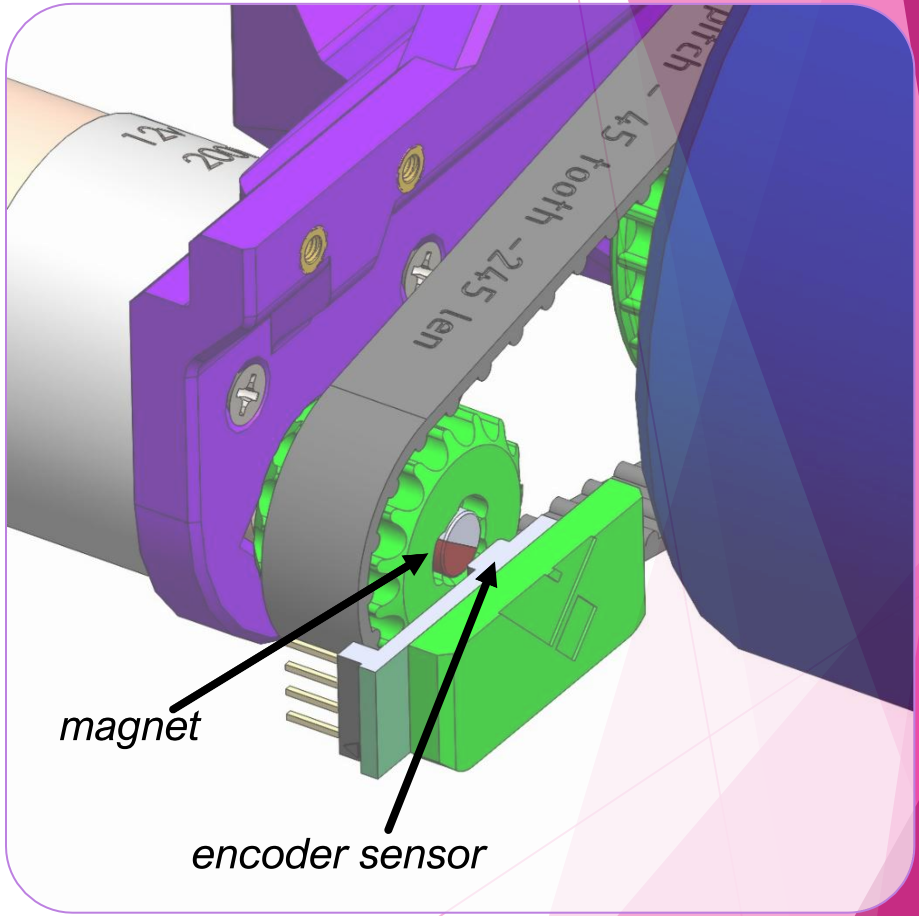
Encoder

The encoder measures the motor rotation by magnetic sensing.

- Readings are reported in 14-bit integers
- Values are always positive, from 0 to 360
- Encoder reports fixed positions and we derive speed from the changes over time
- Two motor turns = One wheel turn



contactless encoder, and magnet

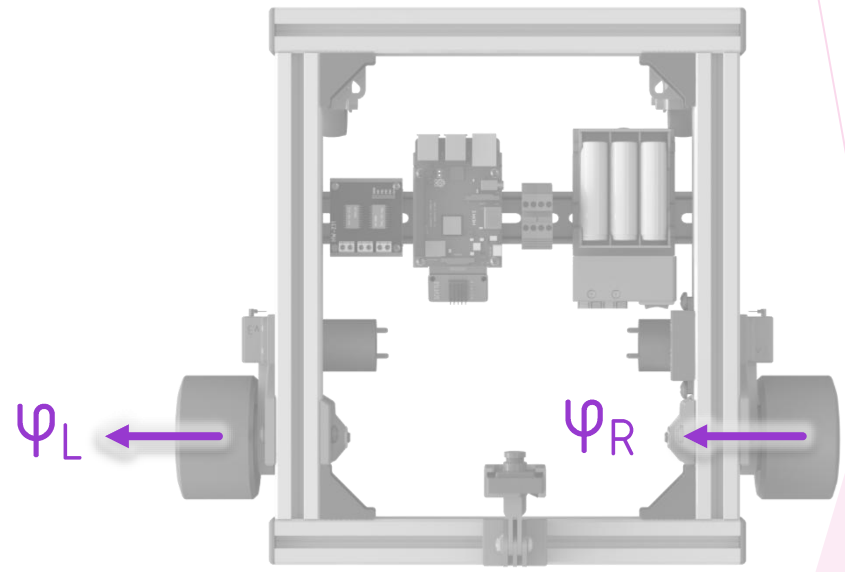


Wheel Rotation

- Phi is the angle of the wheel.
  - Use the right-hand-method to identify the forward direction for phi
  - Let ‘a’ be the encoder reading (from 0 to 360)
  - Let ‘N’ be the small:large pulley ratio
  - Then,

$$\varphi_R = a_R * N$$

$$\varphi_L = (360 - a_L\ ) * N$$



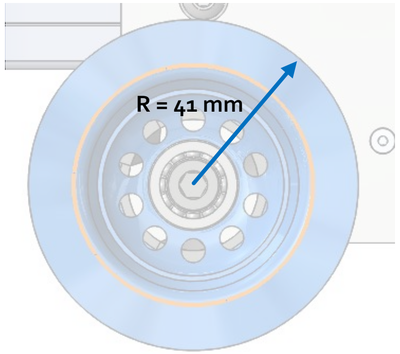
SCUTTLE wheel rotation vectors, phi-left + phi-right

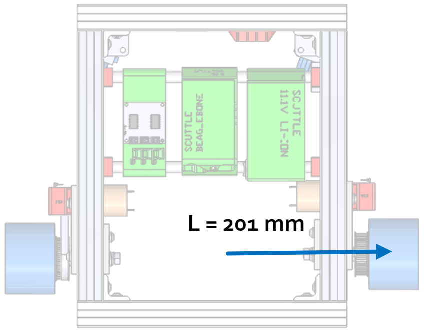
Inverse Kinematics

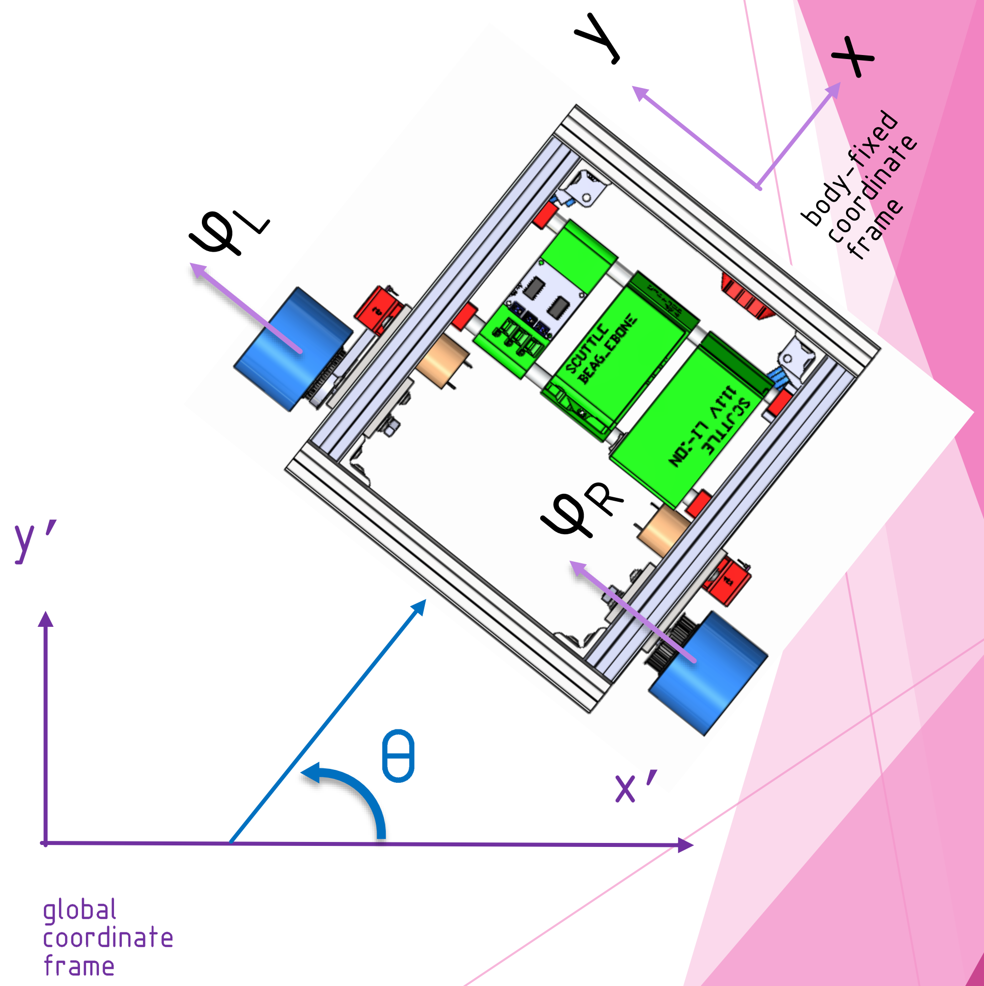
- Inverse Kinematic equation:
  - Input the desired speed and angular speed, and output the left and right wheel speeds.
  - These equations are written in the robot-fixed frame

$$\begin{bmatrix}\dot{\varphi}_L\\ \dot{\varphi}_R\end{bmatrix} = \begin{bmatrix}1/R & -L/R\\ 1/R & L/R\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix}$$

matrix multiplication: $[C] = [A][B]$







Robot Frame

Forward and inverse kinematics in the robot frame:

"Kinematics"

Use the **wheel** speeds to obtain the **chassis** speeds

$$\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ -R/2L & R/2L \end{bmatrix}\begin{bmatrix}\dot{\varphi}_L\\ \dot{\varphi}_R\end{bmatrix}$$

Solve for

"Inverse Kinematics"

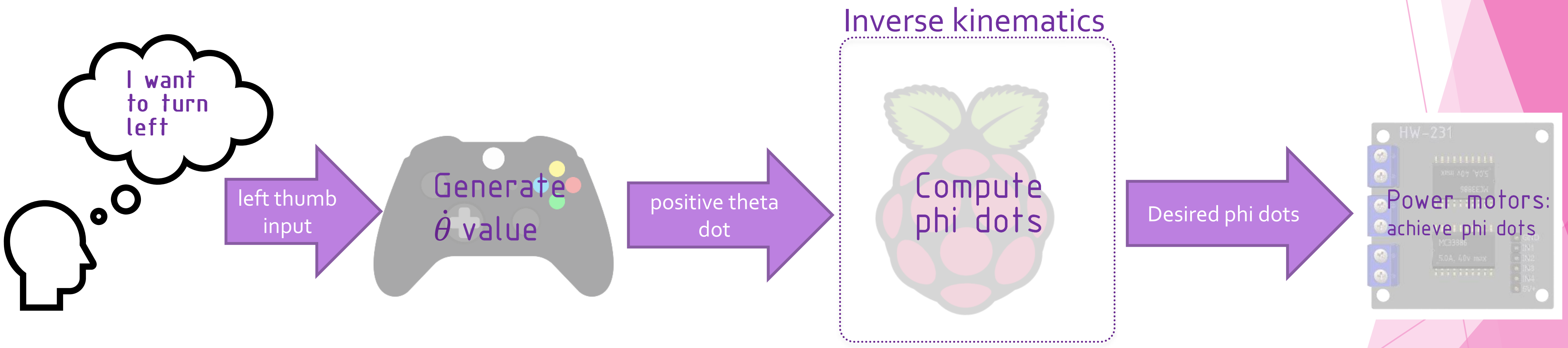
Use the **chassis** speeds to obtain the **wheel** speeds

$$\begin{bmatrix}\dot{\varphi}_L\\ \dot{\varphi}_R\end{bmatrix} = \begin{bmatrix} 1/R & -L/R \\ 1/R & L/R \end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix}$$

Solve for

Use Case

How do we use inverse kinematics?



Global Frame

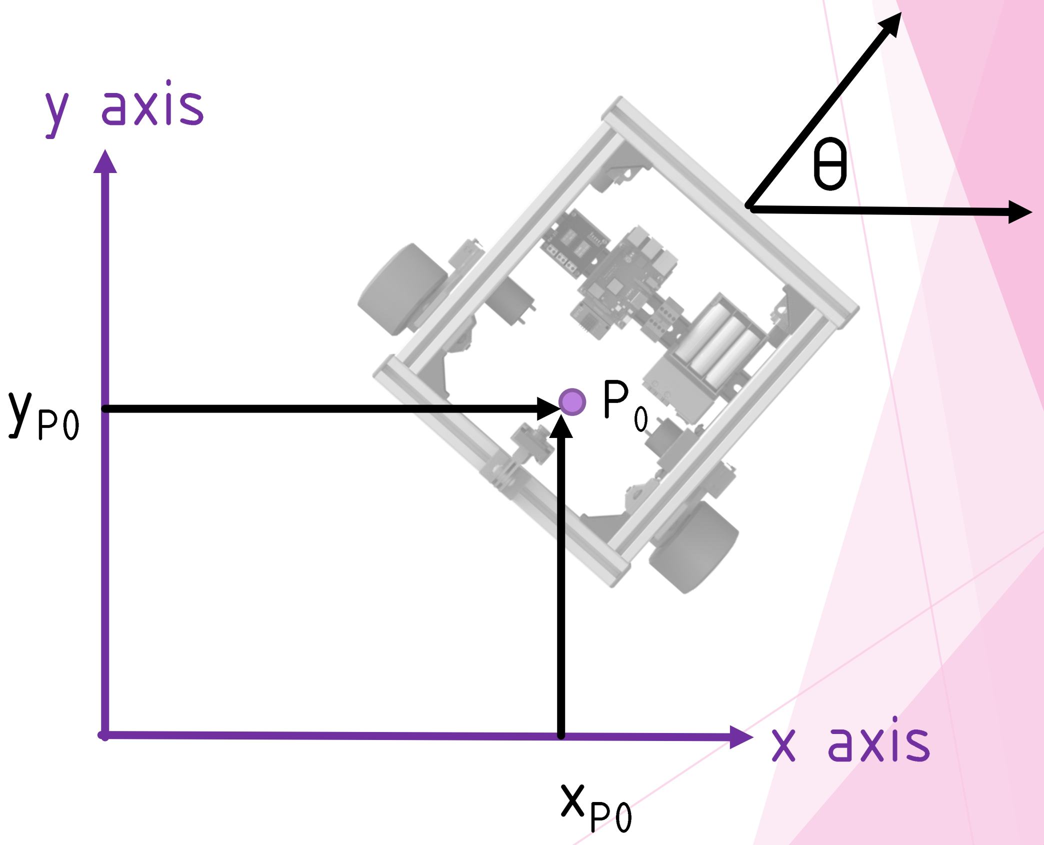
Coordinate Frames

Describing the robot in the inertial (global) frame:

- The global frame is selected by the user and fixed to the earth. It maybe the corner of a room.
- From the global frame we can measure inertia, so it may be called the inertial frame.
- P0 is the robot location, in the global frame.
- P0 is defined by the center of the wheelbase.

$$q^I = \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix}$$

global coordinate frame

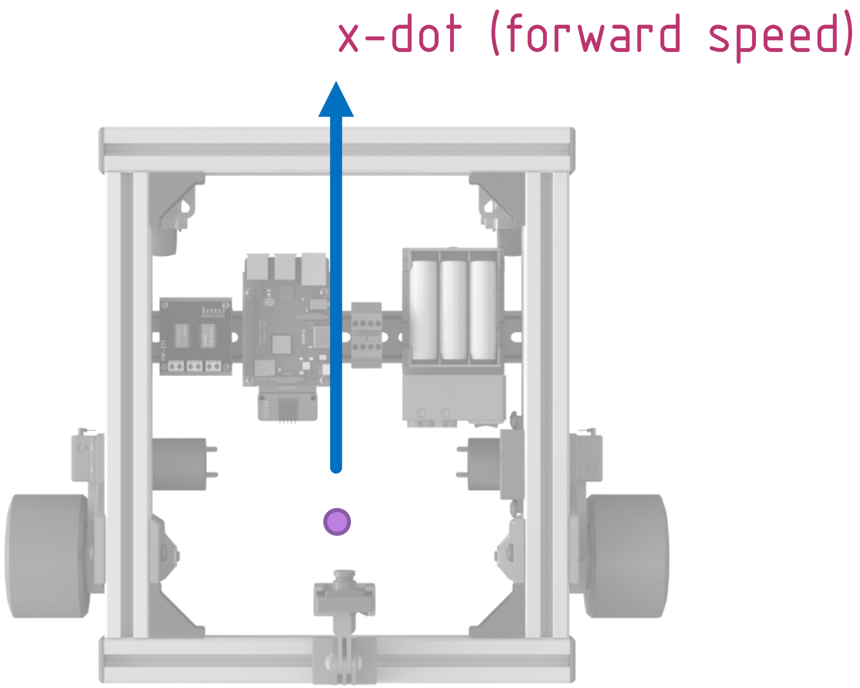


Body-Fixed Frame

Coordinate Frames

The body-fixed-frame is the coordinate system fixed to the robot.

- the body-fixed frame is used to describe obstacles or room locations.
- the robot's own location in this frame is zero for x, y, and theta.
- if the robot gains a module, it is useful to define in the body-fixed-frame.



x-dot may be positive or negative

Kinematics for ROS

ROS (robot operating system) offers documentation with some standard methods for robotic motion and the components thereof. Motion is described using a twist message to command an actuator. Think of twist as a combination of translation and rotation.

Within ROS documentation, this topic falls under robotic navigation or path planning.

https://wiki.ros.org/navigation/Tutorials

What is a twist message?

"[Twist.msg] expresses velocity in free space broken into its linear and angular parts"

-ROS documentation

Motion Variables

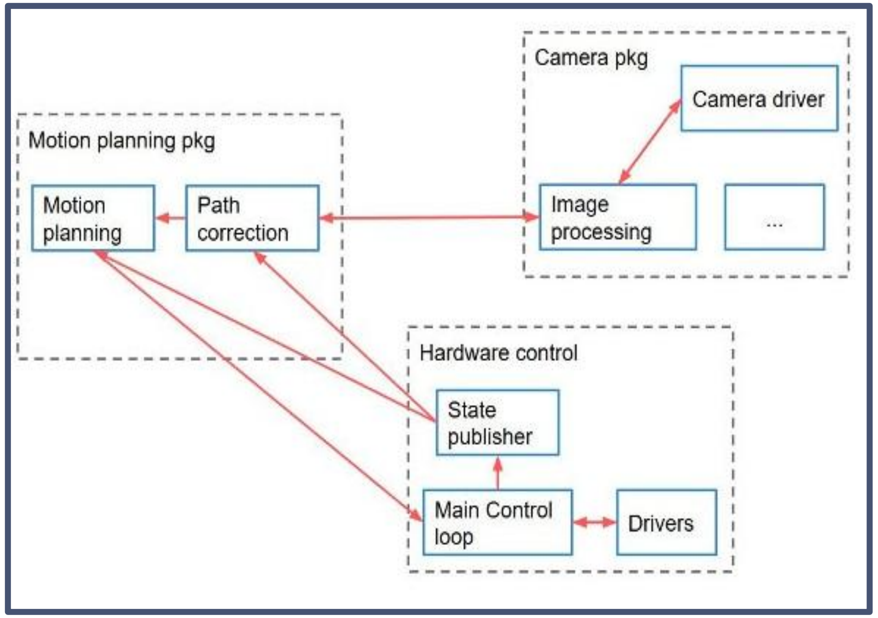
ROS	SCUTTLE
speed.linear.x	x_dot or xd
speed.linear.y	always zero
speed.linear.z	always zero
speed.angular.x	always zero
speed.angular.y	always zero
speed.angular.z	theta_dot or td

How to relate ROS twist components with SCUTTLE?

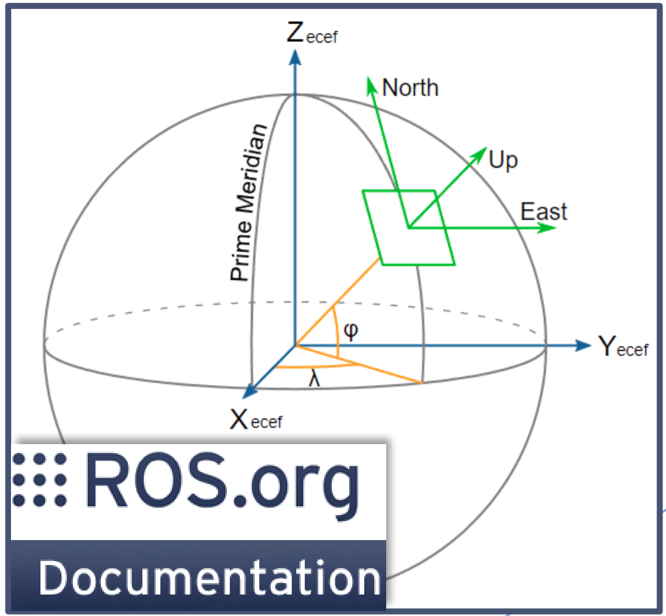
Kinematics for ROS

- First diagram shows how the ROS node (blue rectangle) lives in a package, and interacts with other packages.

What is a ROS node?
Simple explanation

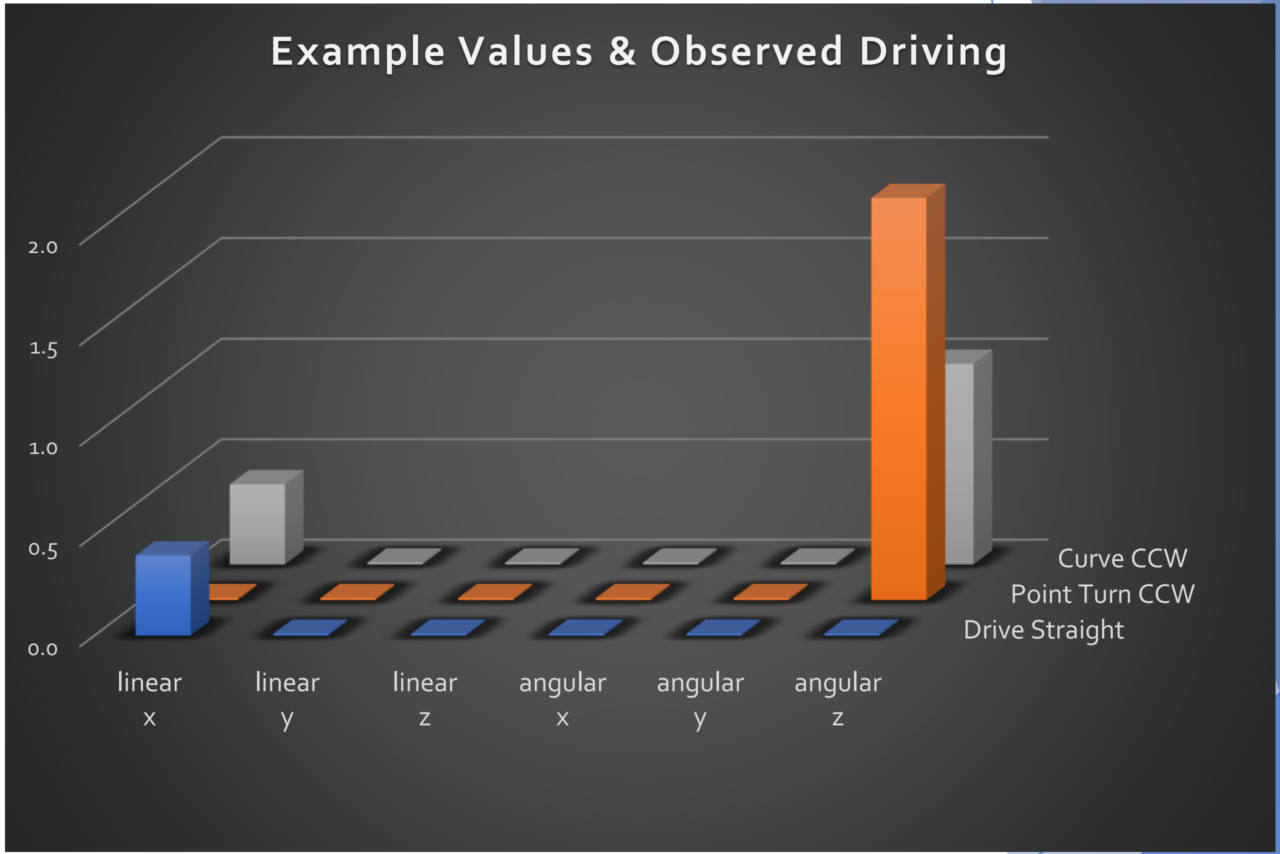


Naming conventions for mobile platforms by ROS.org



Speeds

- What values should I target for motion? The chart shows peak speeds for the standard SCUTTLE with 12v battery.
- Rotation can be made at 0.75 radians per second while driving 0.3 meters per second forward
- Rotation can be made above 1.5 radians per second while forward speed is zero (a point turn).



- Twist.msg elements above, common (achievable) values to move SCUTTLE

ROS Standards:

- Use the units below in your commands for motion, inside the twist.msg
- Gather more information in the REP archives of ros.org

Units

ROS	SCUTTLE
speed.linear.x	meters per second
speed.angular.z	radians per second

What units are used for the twist message?

Reference

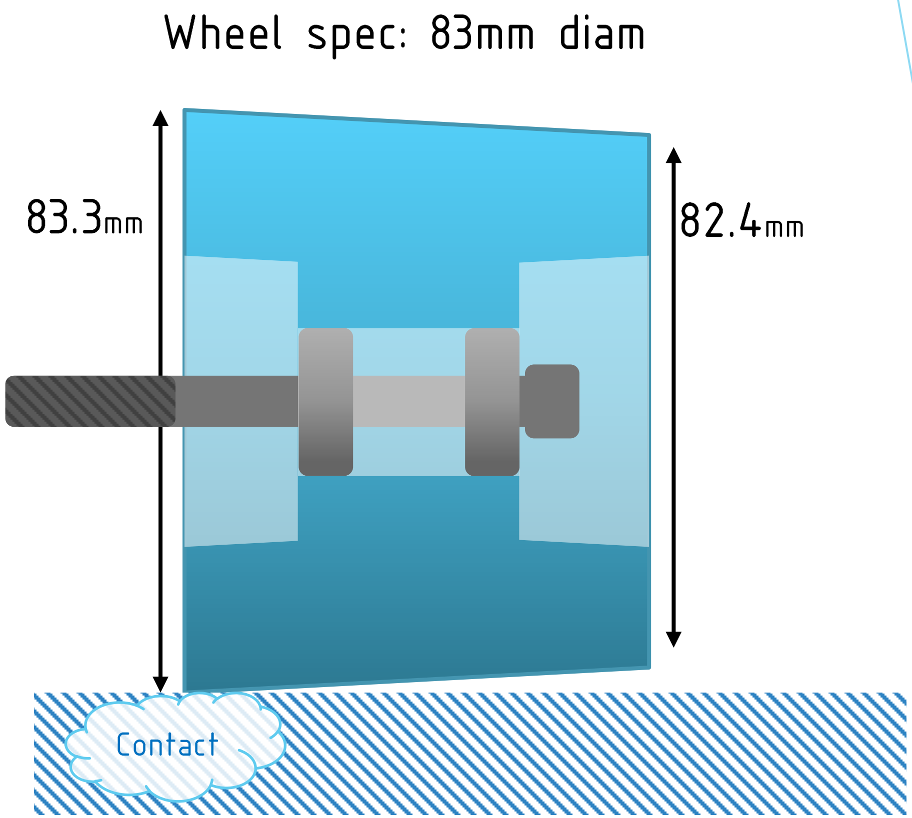
REP:	103
Title:	Standard Units of Measure and Coordinate Conventions
Author:	Tully Foote, Mike Purvis
Status:	Active
Type:	Informational
Content-Type:	text/x-rst
Created:	07-Oct-2010
Post-History:	31-Dec-2014

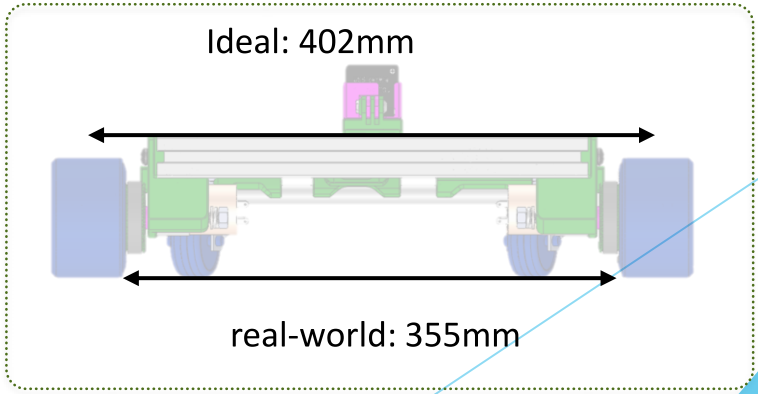
https://www.ros.org/reps/rep-0103.html

All recommended units by ROS.org

Wheelbase

1) The true wheelbase is determined by the contact patch of the wheels on the ground.
   - wheels have a *draw angle* from the manufacturing mold, resulting in a larger inner diameter.
   - Then, the contact patch may lie closer to the robot center, especially with no payload.
   - The inner wheel edge gives a 355mm wheelbase.
   - The wheelbase may change when a heavy load compresses the urethane.





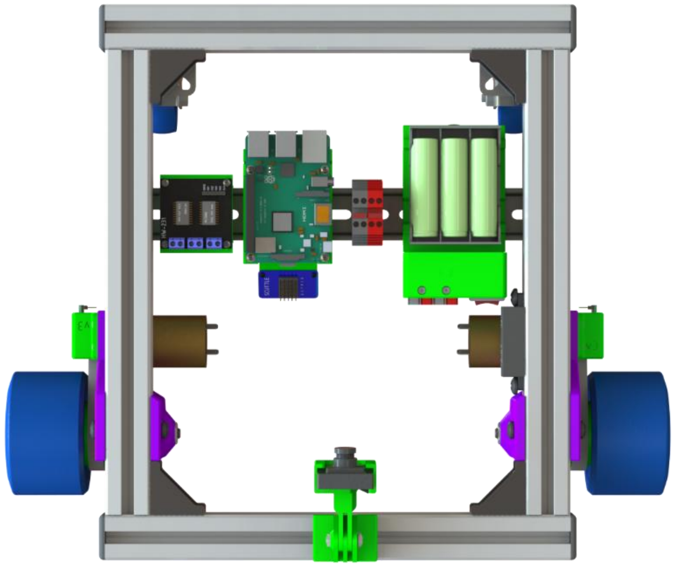
Wheelbase

Kinematics Sources:

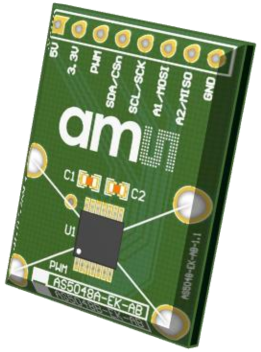
1. Dhaouadi, Rached, and A. Abu Hatab. "Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework." *Advances in Robotics & Automation* 2.2 (2013): 1-7.
2. Hur, Byul, et al. "Open-source Embedded Linux Mobile Robot Platform for Mechatronics Engineering and IoT Education." *Journal of Management & Engineering Integration* 13.2 (2020): 34-44.
3. Achmad, MS Hendriyawan, et al. "ROS-based 2-D Mapping Using Non-holonomic Differential Mobile Robot." *JURNAL INFOTEL* 10.2 (2018): 75-82.

Images

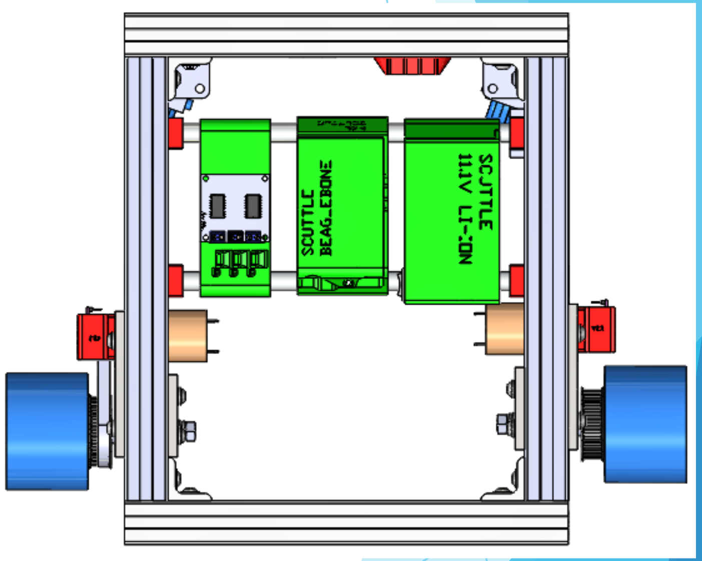
- Use these images for slides & plans



SCUTTLE version 3.0 with Raspberry Pi



encoder, AS5048B



SCUTTLE version 2.4