SCUTTLE Robot Kinematics Guide

revised 2022.04.19

SCUTTLE Kinematics

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

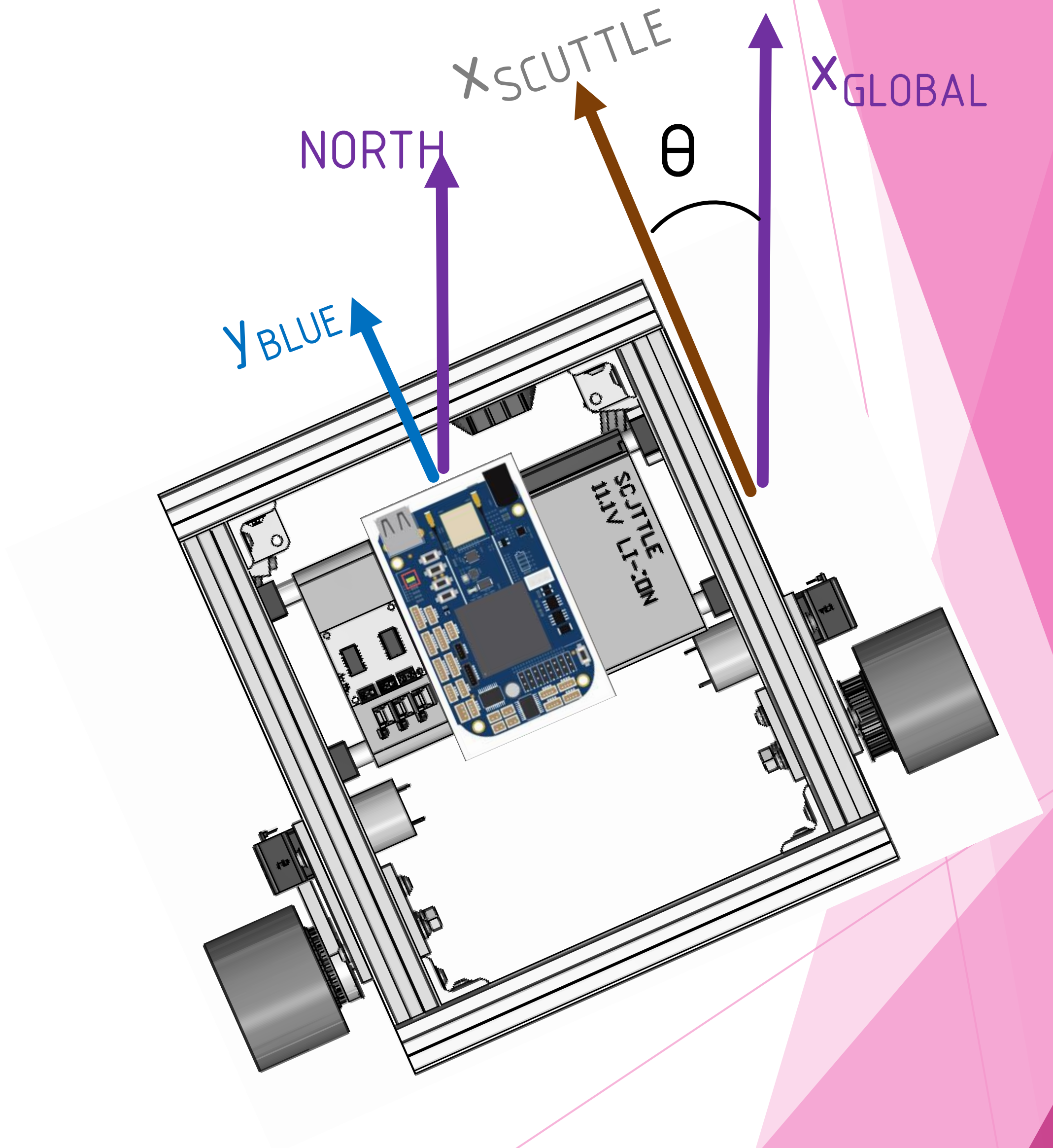
SCUTTLE Coordinates

<u>This slide is for reference.</u>

The beagle circuit board has an IMU with y-axis pointing along the USB port.

The global x- is often decided to be aligned with magnetic North.

The SCUTTLE body-fixed X vector is aligned with the forward direction.



SCUTTLE: a Non-Holonomic System

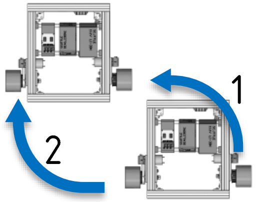
A holonomic robot has the same number (or more) of controllable degrees of freedom as the number of degrees of freedom.

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

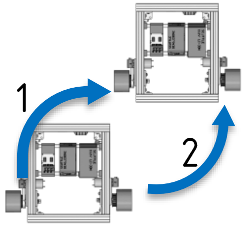
In a non-holonomic system, the final position of the robot depends on the path taken to achieve the movement.

Example:

Move right wheel, then left

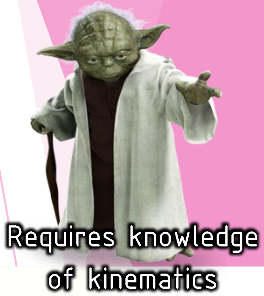


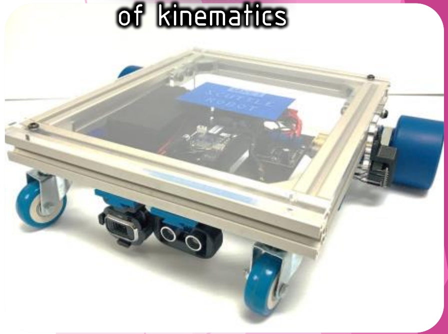
Move left wheel, then Right



Holonomic:

Easier navigation
More parts
Less robust



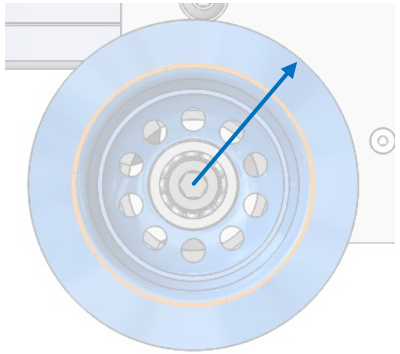


Non-Holonomic:

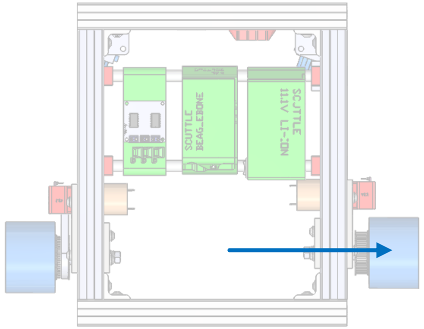
Harder navigation
Fewer parts
More robust

SCUTTLE Kinematics

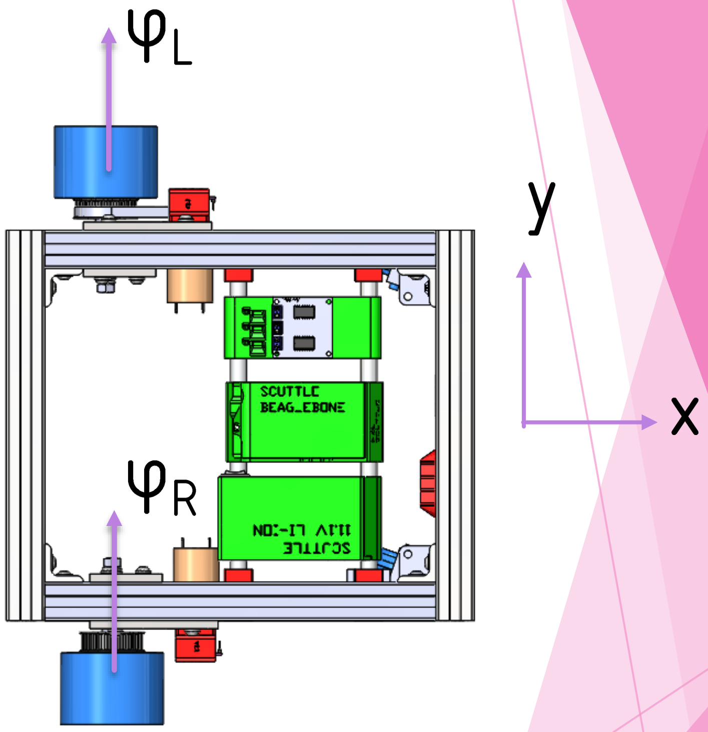
- The **Chassis Geometry** determines the equations for kinematics.
- The **radius**, r, is the radius of the driven wheel
- The **half-wheelbase**[(1)], L, is the space from wheel center to center divided in two



R = 41 mm



L = 201 mm [(1)]



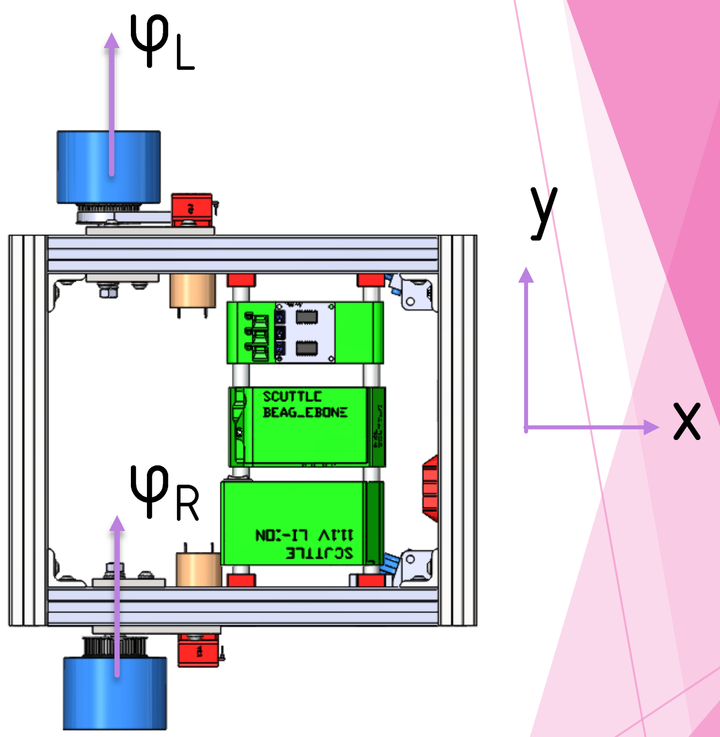
SCUTTLE Kinematics

- 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

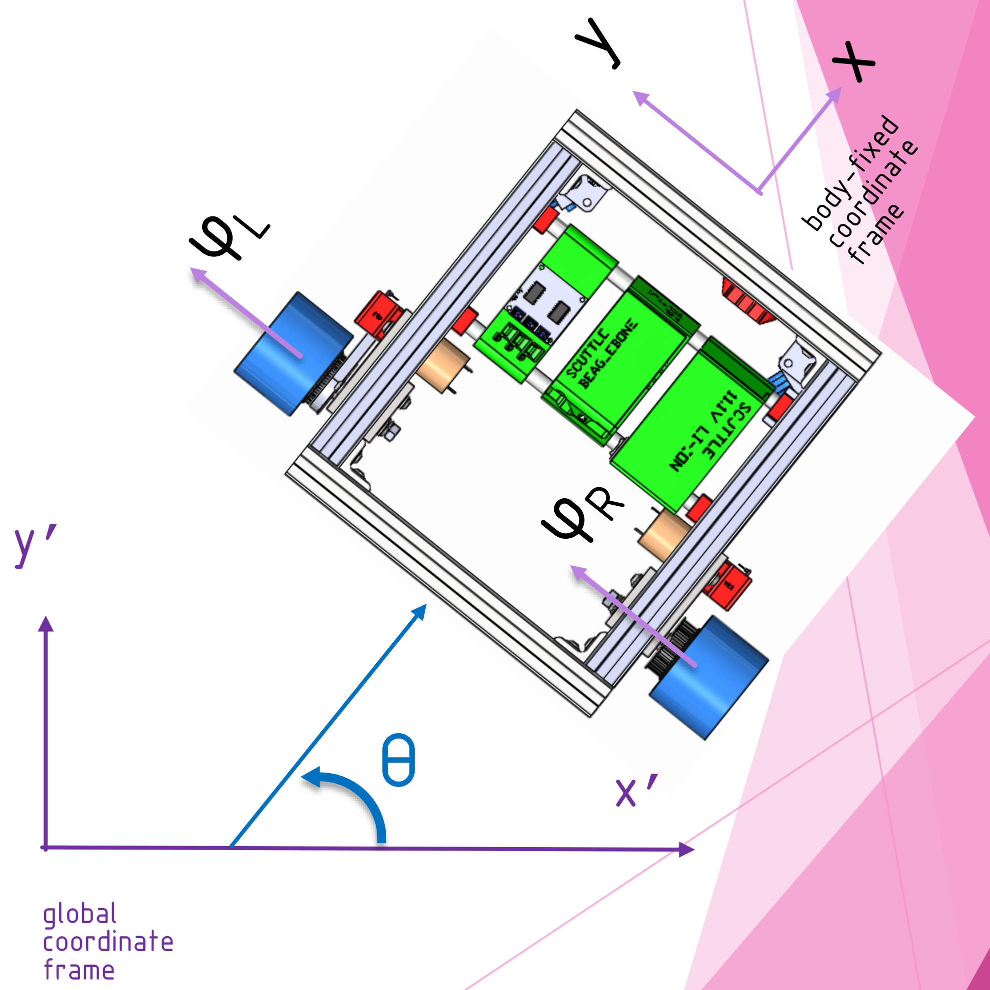
```
phiDots = np.array( [ pdl, pdr ] ) # python syntax
```



SCUTTLE Kinematics

- **Theta** describes the difference between the body-fixed frame and the global frame.
- The **rotation matrix** converts body-fixed coordinates to the global coordinates

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

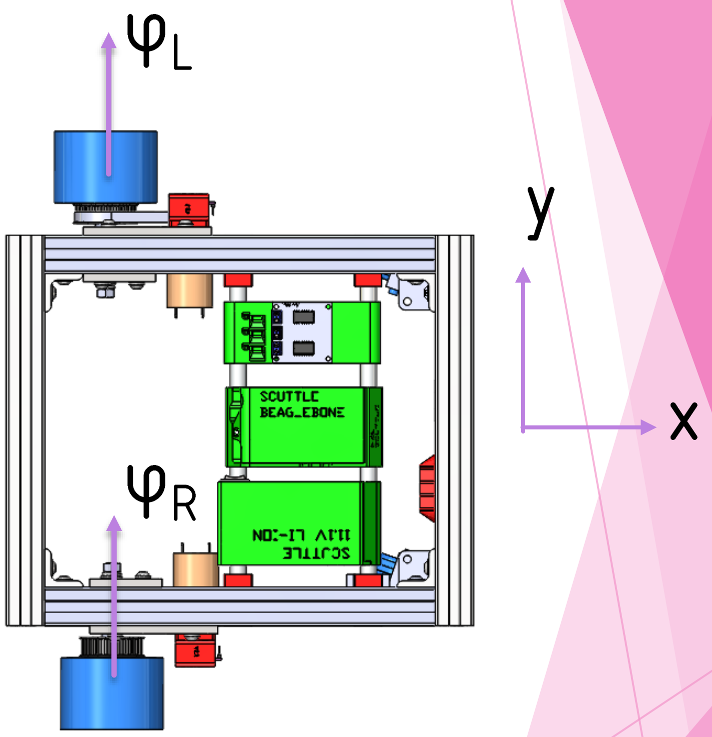


SCUTTLE Kinematics

- 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

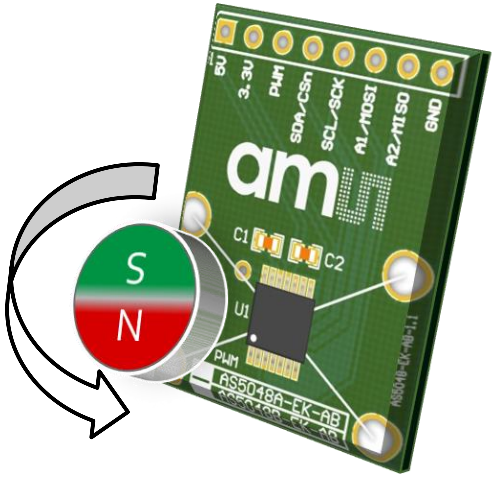
- The Kinematic Equation generates chassis motion information.
  - input the wheel speeds and output the (translational and rotational) chassis speeds

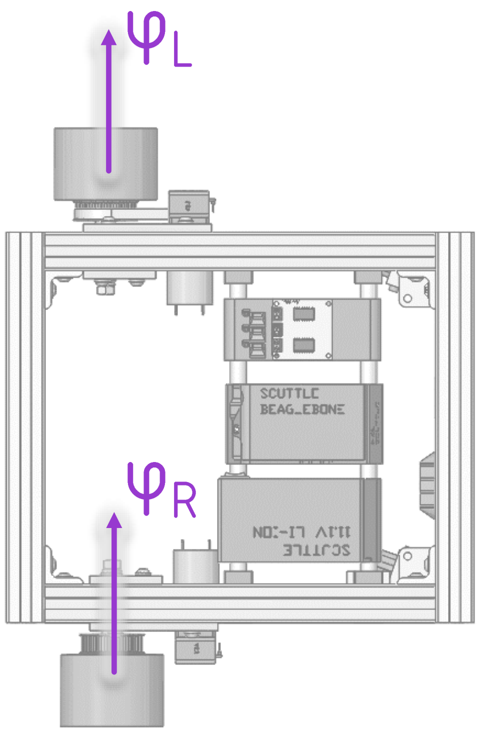


$$\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}R/2 & R/2\\ -R/2L & R/2L\end{bmatrix}\begin{bmatrix}\dot{\varphi}_L\\ \dot{\varphi}_R\end{bmatrix}$$

SCUTTLE Kinematics – Encoder

- Phi is the angle of the wheel.
  - Let 'a' be the encoder reading (from 0 to 360)
  - Let 'N' be the small:large pulley ratio
  - Then,
    - $\varphi_R = a_R * N$
    - $\varphi_L = (360 - a_L) * N$



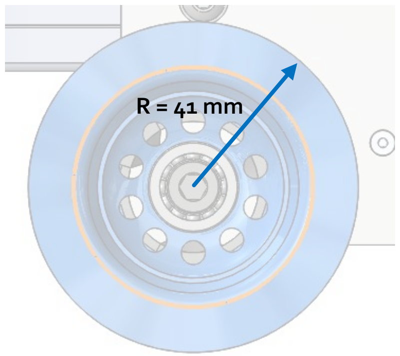


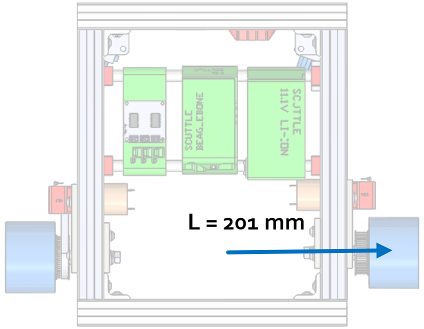
SCUTTLE Kinematics

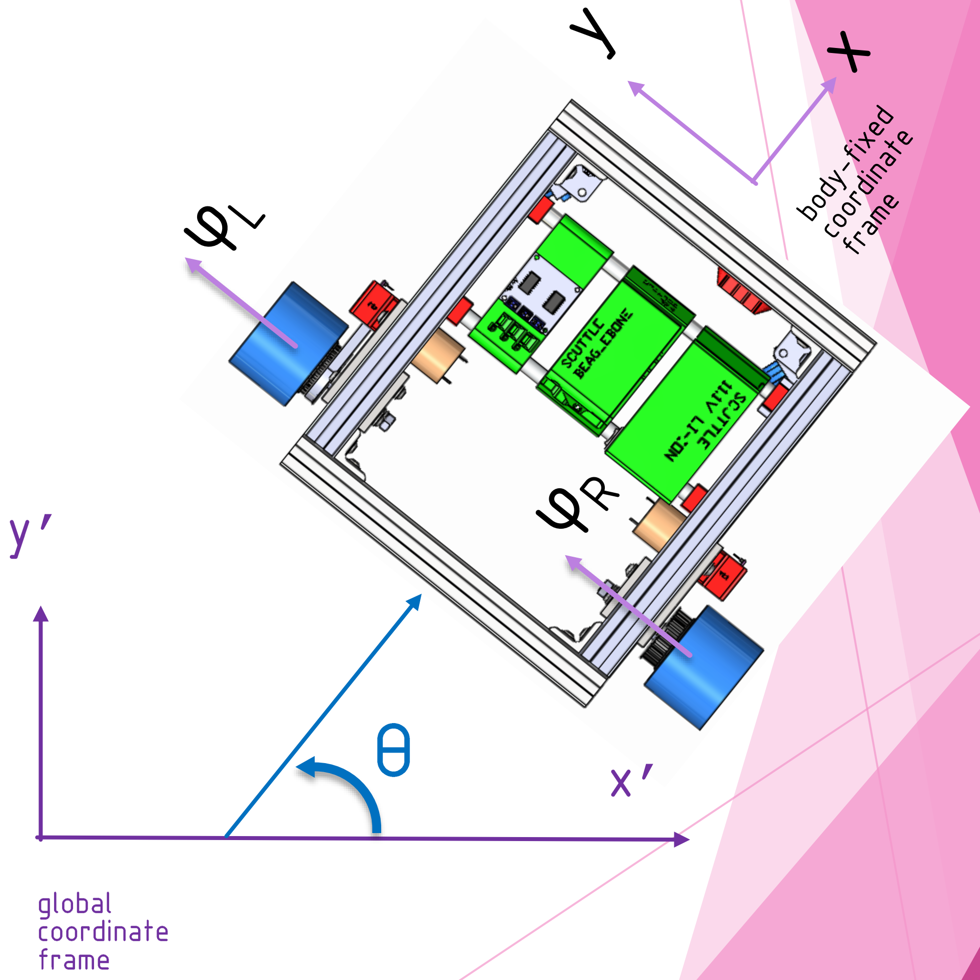
- Inverse Kinematic equation:
  - Input the desired speed and angular speed, and output the left and right wheel speeds.
  - These equations are written in the robot-fixed frame

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

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







SCUTTLE Kinematics

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"

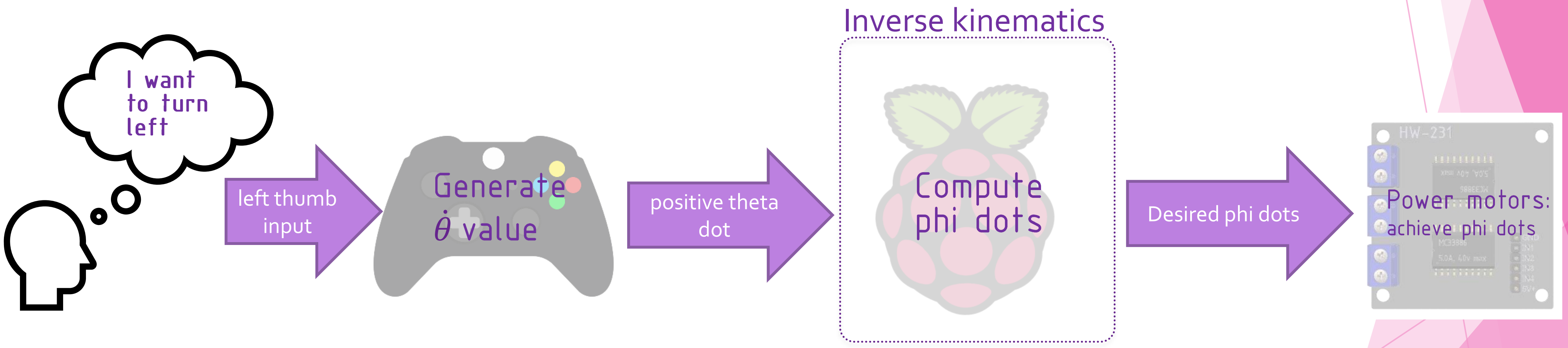
Use the **chassis** speeds to obtain the **wheel** speeds

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

Solve for

SCUTTLE Kinematics

How do we use inverse kinematics?



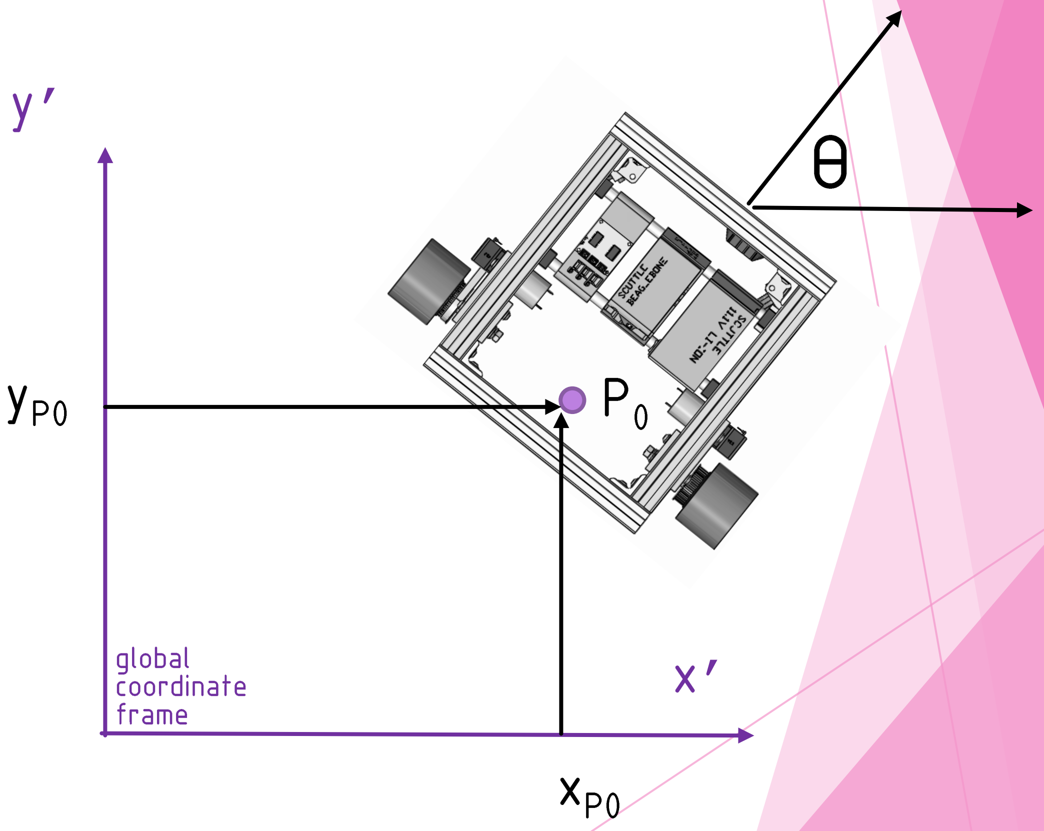
SCUTTLE Kinematics (prt 2)

This section will be expanded to discuss navigation & inertial frame

Describing the robot in the inertial (global) frame:

- P0 describes the location of the robot and is defined by the center of the wheelbase.

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



Kinematics for ROS

What is a twist message?

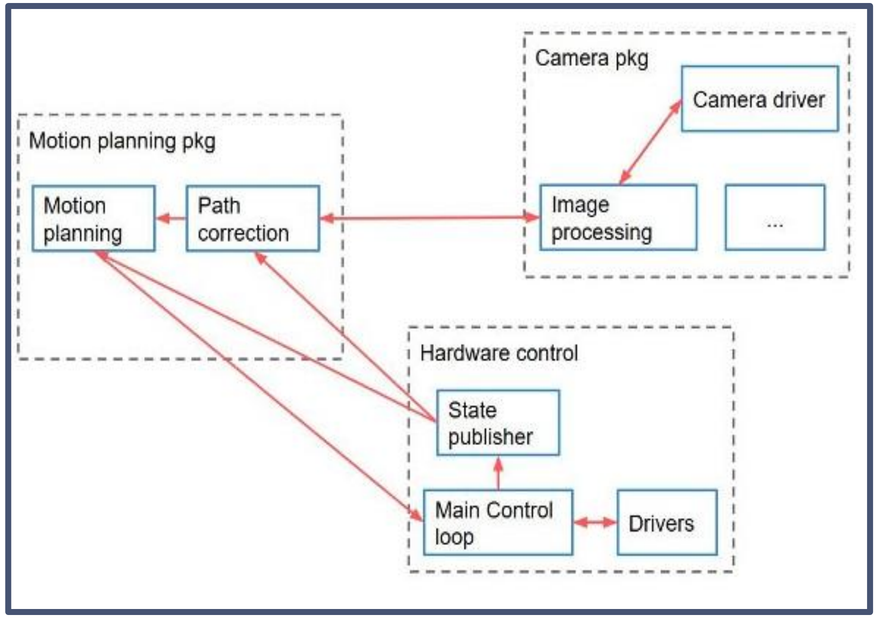
"[Twist.msg] expresses velocity in free space broken into its linear and angular parts"

-ROS documentation

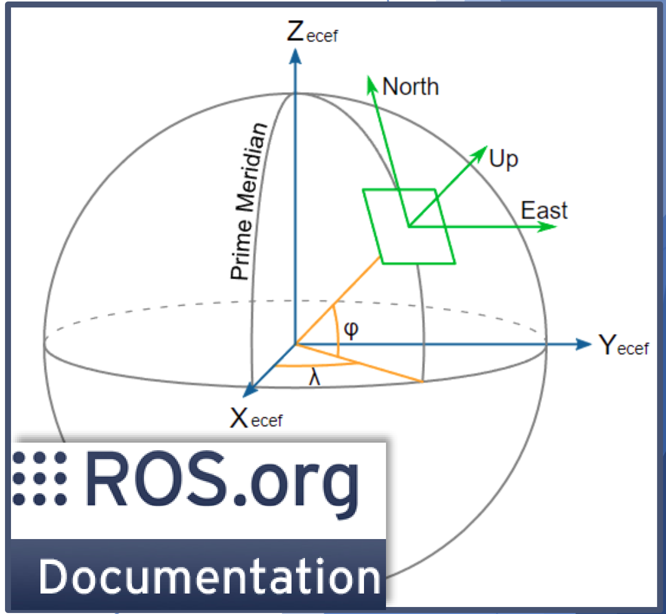
How to relate ROS twist components with SCUTTLE?

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

What is a ROS node? Simple explanation

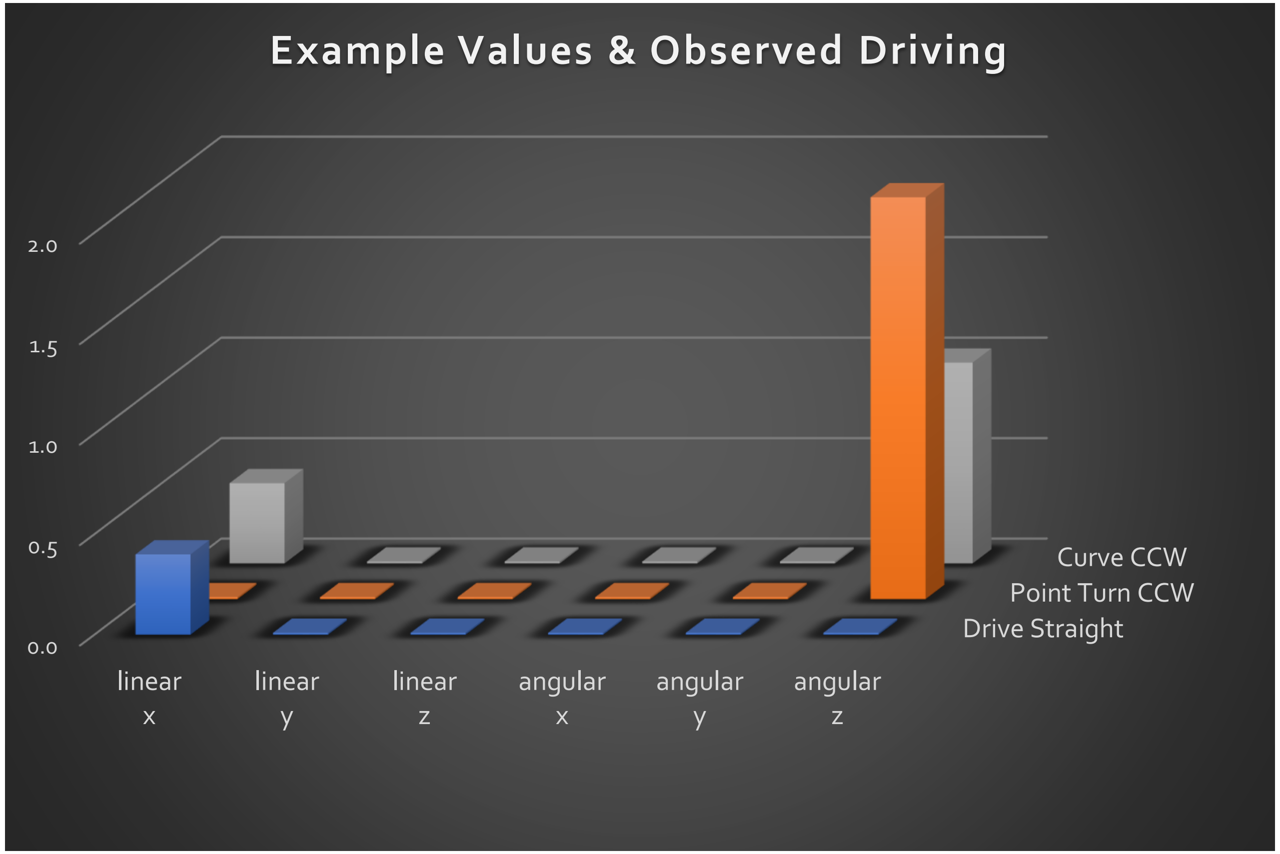


Naming conventions for mobile platforms by ROS.org



Kinematics for ROS

- Twist.msg elements below are common (achievable) values to move SCUTTLE



What units are used for the twist message?

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

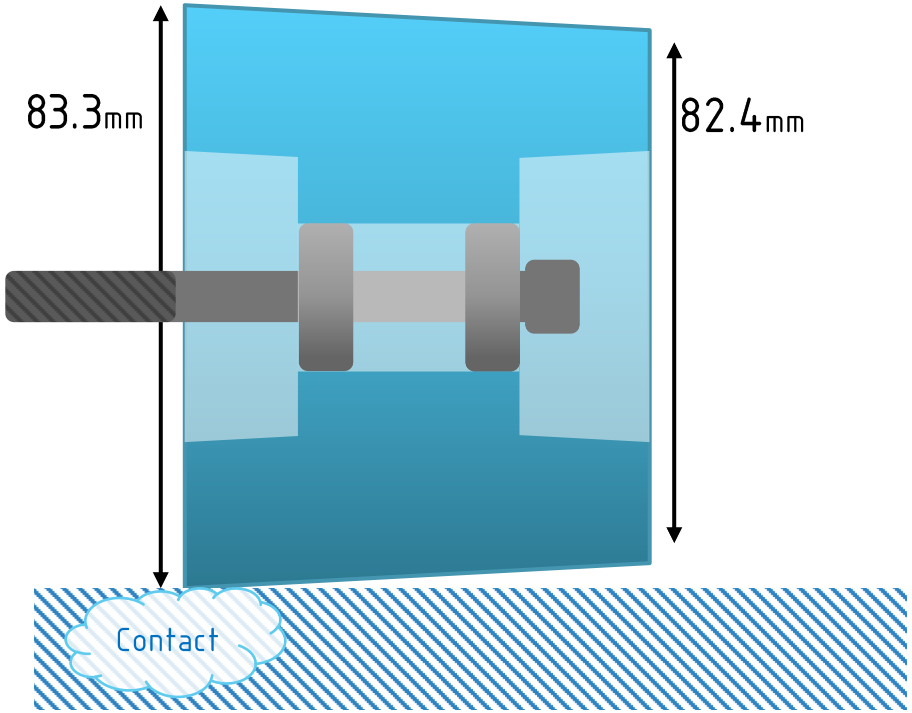
All recommended units by ROS.org

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

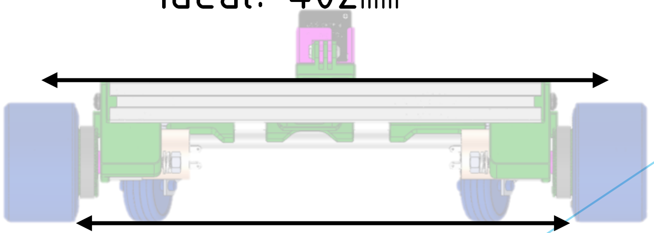
Kinematics Footnotes

1) The true wheelbase is determined by the contact patch of the wheels on the ground.
   - Most 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.

Wheel spec: 83mm diam



Wheelbase



Recommended reading:

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.