Lecture 11

Kinematics of Wheeled Mobile Robots

CS 3630

Logistics Robots

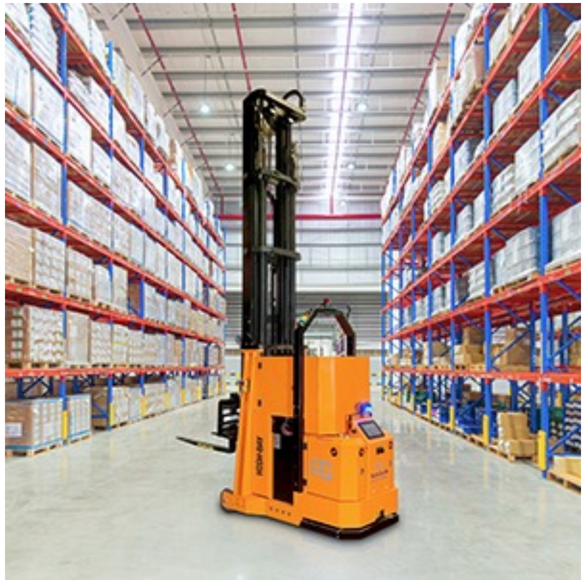
What exactly is logistics?

- Logistics is the management of the flow of things from point of origin to point of consumption [Wikipedia].
- This typically involves multiple stages of packaging, routing, transport.
- There are plenty of robotics applications:
    - Loading/unloading
    - Palletizing
    - Cargo container transport
    - Packaging
    - Last mile delivery
    - Warehouse operations

➢ For now, we will consider the narrow problem of warehouse operations, in particular, the problem of moving inventory from Point A to Point B in a warehouse.

Robots in Warehouses

A Few Warehouse Robots



Autoguide Mobile Robots

GreyOrange Inc.

Tompkins Robotics

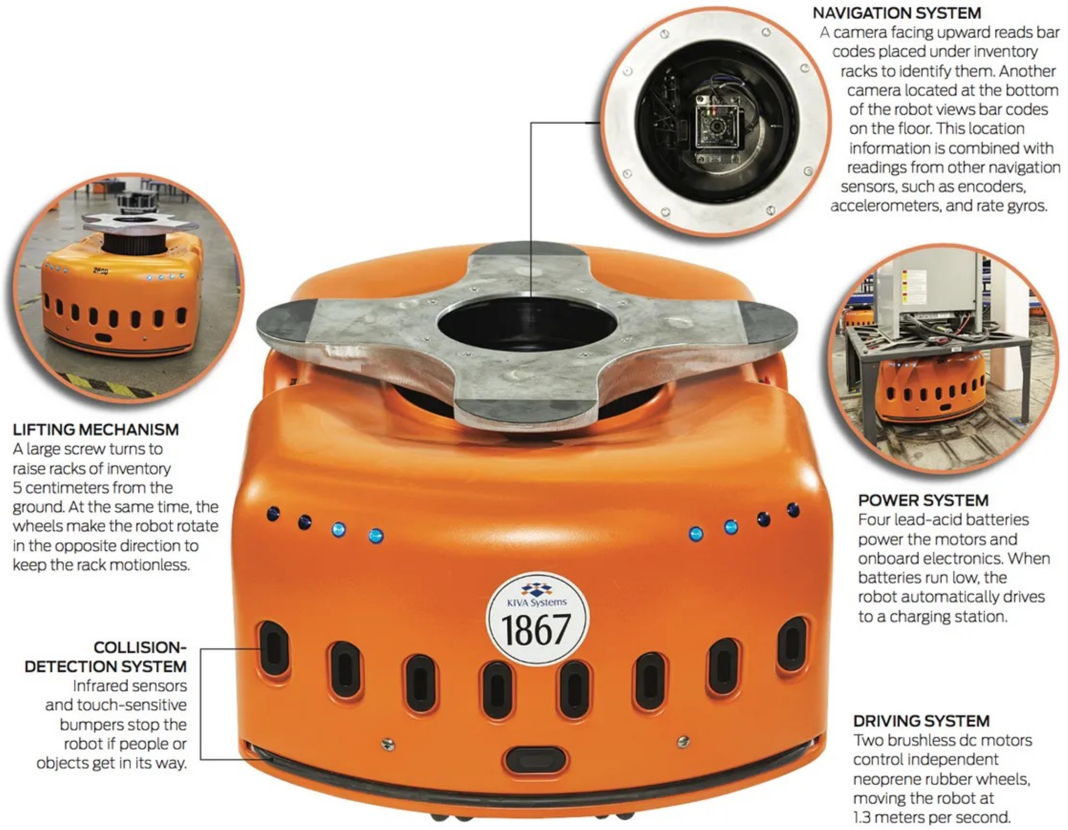
Milvus Robotics

A few mobile robots whose purpose in life is to move inventory from place to place in large warehouses.

From Kiva to Amazon Robotics

- 2003: Kiva Systems founded
- 2009: Rank #6 in Inc. 500 list of fastest growing co's in America
- 2012: Acquired by Amazon for $775M
- 2015: Name change: Amazon Robotics LLC
- 2019: *More than 200,000 robots deployed in Amazon warehouses*

Peter Wurman, Mick Mountz, Raff D'Andrea



IEEE Spectrum, Jul 2008

Amazon’s Warehouse Robots

Fetch Robotics

- Cloud robotics platform (claim to be the first)
- Mobile manipulation
- Sponsored competition at ICRA (GT won, and took home a shiny new robot).
- 2014: Founded (after Willow Garage ended)
- 2019: AI Breakthrough Award (best overall robotics company)
- 2021: Acquired by Zebra for $305M

CEO, Melonee Wise

Autonomous Mobile Robots

In the world of warehouse robotics, we there are two main categories or mobile robot platform:

- Automated Guided Vehicles (AGVs)
  - Follow fixed routes
  - Rely on wires or magnets embedded in the floor to track routes
  - Simple sensing to avoid collisions (typically, simply stop when an obstacle appears)
  - Rely on predictable and known environment
  - Train the humans to avoid the robots

- Autonomous Mobile Robots (AMRs)
  - Capable of planning general motion
  - Typically require a map of the environment
  - Can navigate based on obstacles (i.e., more than simple collision avoidance)
  - Robots know how to avoid the humans

Actions

Until this point, we have ignored the issues related to robot motion:

- The trash sorting robot had built-in sorting actions.
- The vacuuming robot had built-in motion primitives to navigate from room to room.
- We modeled uncertainty, but we really didn't do any work to develop these models, which should be related to reliability of the robot's actions/motions.

Now, we'll take a first look at robot motion:

- Rolling wheels induce motion of a mobile platform.
- Uncertainty in the effects of actions is modeled directly in terms of the robot's motion.

➢ We'll start with the kinematics of omni wheels, then extend to Differential-Drive Robots (DDRs)

Omni Wheels

Typical wheel:

- Rolls forward (the driving direction) without slipping
- Cannot slide perpendicular to the steering direction
- Wheel velocity is therefore always in the driving direction
- The inability to slide is a nonholonomic constraint

Omni wheel:

- Rolls forward (the driving direction) without slipping
- Can slide perpendicular to the steering direction
- ***Wheel velocity not constrained to be in the driving direction!***
- Sliding is passive, just the right amount to accommodate the wheel velocity.

Typical Omni-Wheel robot

The reason for three wheels:

- Steering directions of the three wheels positively spans the plane, plus stability.
- Can move in any direction instantaneously by an appropriate choice of wheel speed.

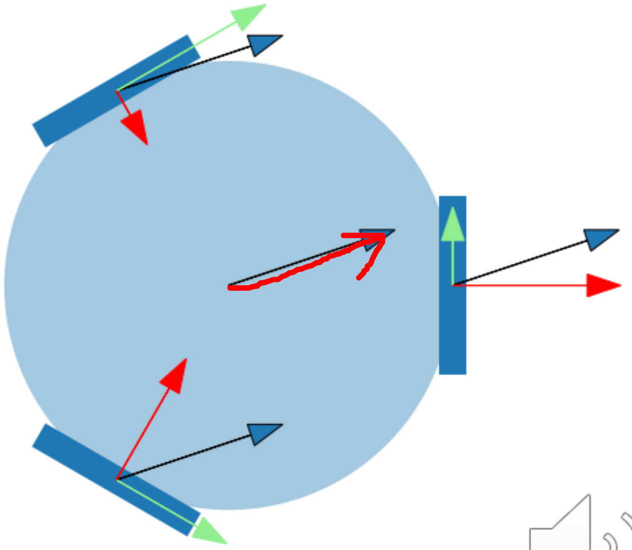
Wheel Kinematics

- We consider only pure translations (we'll consider orientation and rotation later).
- If the robot moves with a pure translational velocity, then every point on the robot moves with the same velocity.
- Define the translational velocity of the robot to be

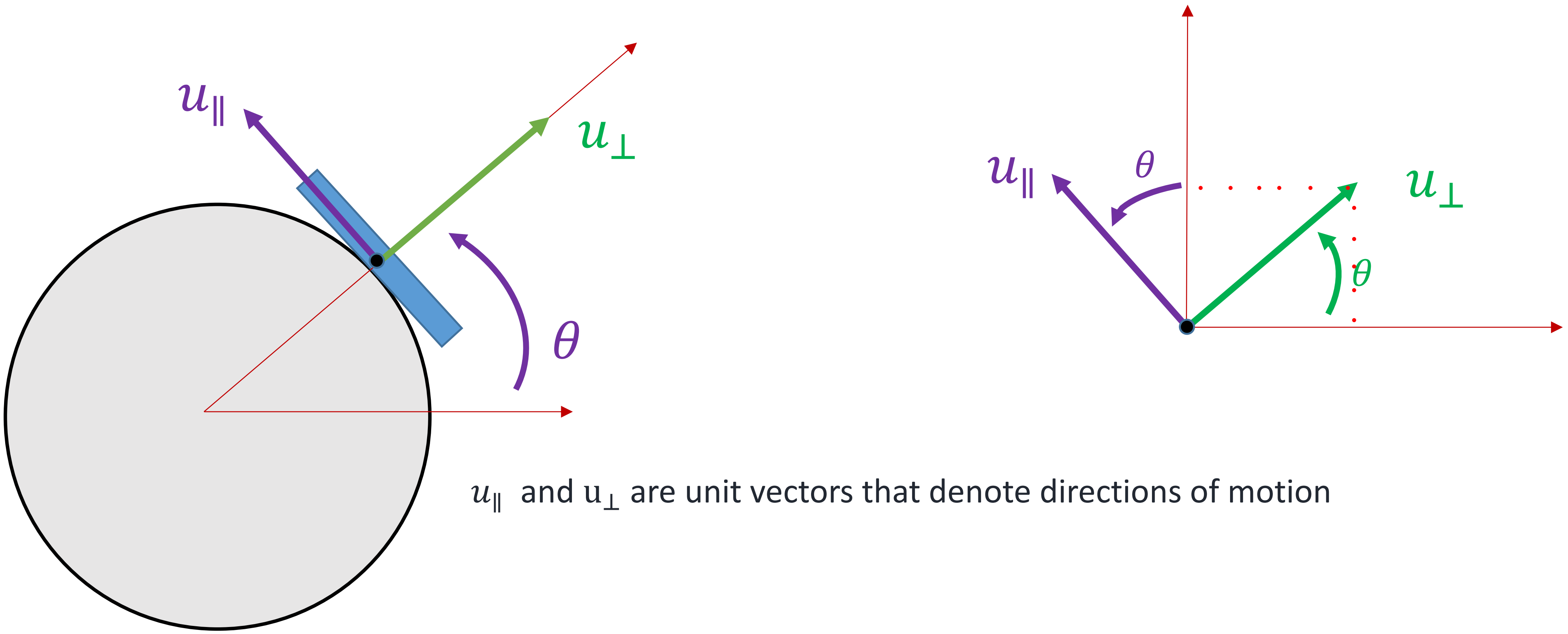
$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

The velocity of each wheel can be decomposed into two components: $v_{\|}$ and $v_{\perp}$.

- $v_{\|}$ is the component of wheel velocity that is parallel to the driving direction.
- $v_{\perp}$ is the component of the wheel velocity that is perpendicular to the driving direction.



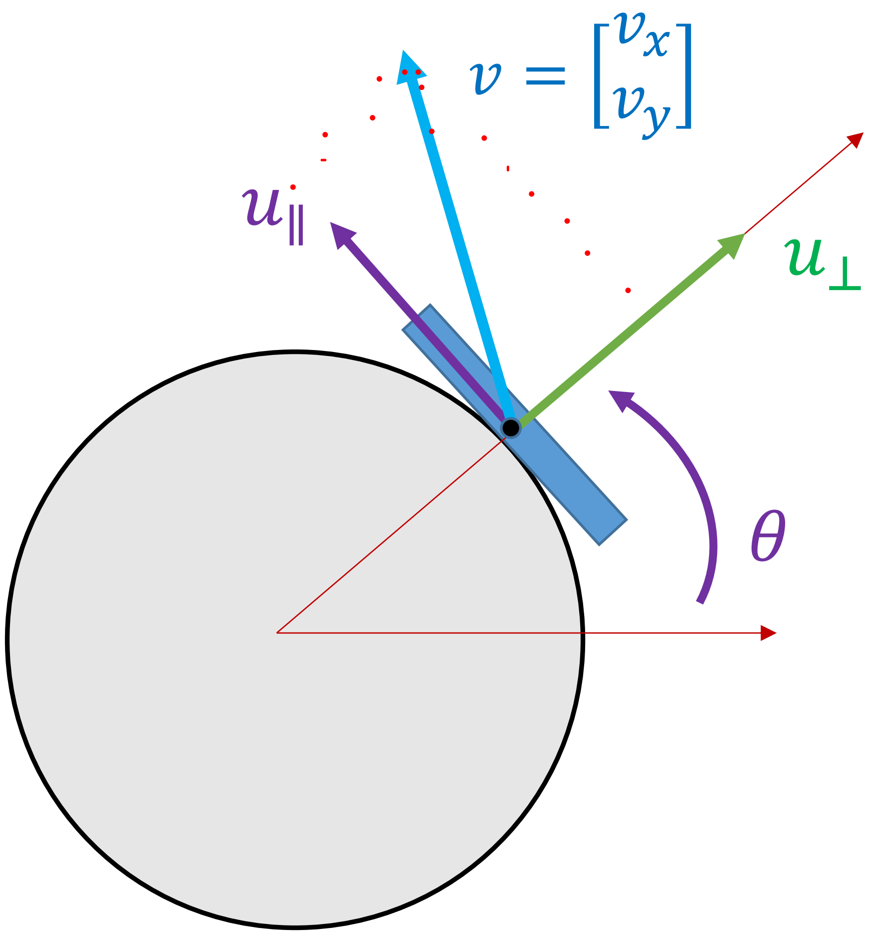
Decomposing Robot Velocity



$u_\parallel$ and $u_\perp$ are unit vectors that denote directions of motion

$$u_\perp = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad u_\parallel = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

Decomposing Robot Velocity



- We can now decompose v into the components parallel to and perpendicular to the steering direction.
- This is done by projecting v onto $u_\parallel$ and $u_\perp$

$$v = (v \cdot u_\parallel)u_\parallel + (v \cdot u_\perp)u_\perp$$

which can be written as

$$v = v_\parallel u_\parallel + v_\perp u_\perp$$

where

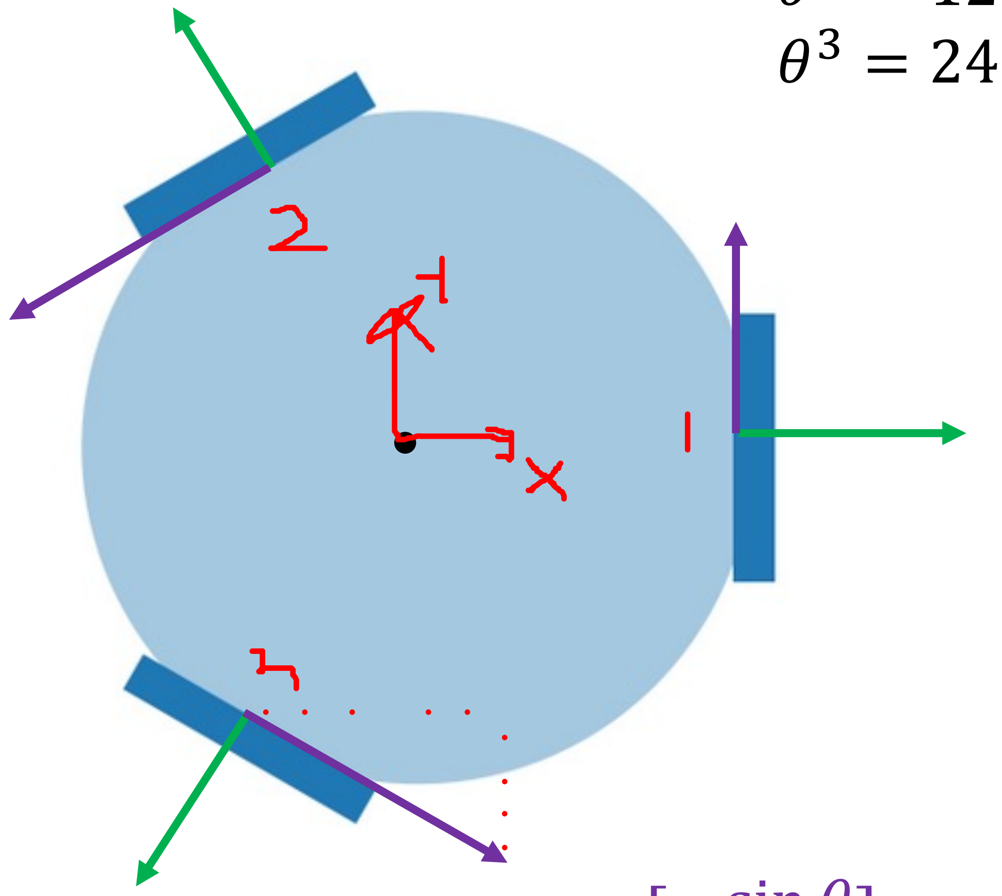
$$v_\parallel = -v_x \sin\theta + v_y \cos\theta$$
$$v_\perp = v_x \cos\theta + v_y \sin\theta$$

$$u_\perp = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad u_\parallel = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

Note that $v_\parallel$ and $v_\perp$ are scalars!

Three Uniformly Positioned Wheels

$$\theta^1 = 0$$
$$\theta^2 = 120$$
$$\theta^3 = 240$$



$$u_{\parallel} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

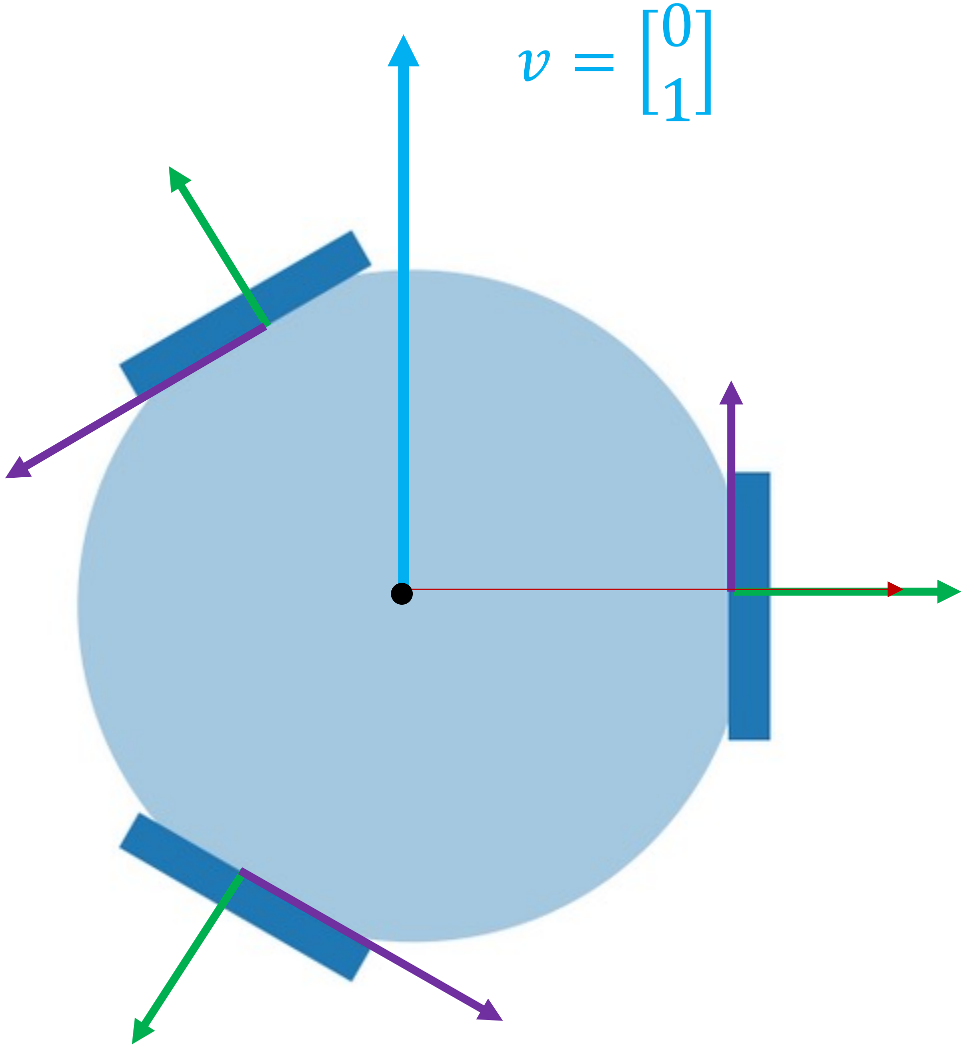
$$u_{\parallel}^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$u_{\parallel}^2 = \begin{bmatrix} -0.8660 \\ -0.5 \end{bmatrix}$$

$$u_{\parallel}^3 = \begin{bmatrix} 0.866 \\ -0.5 \end{bmatrix}$$

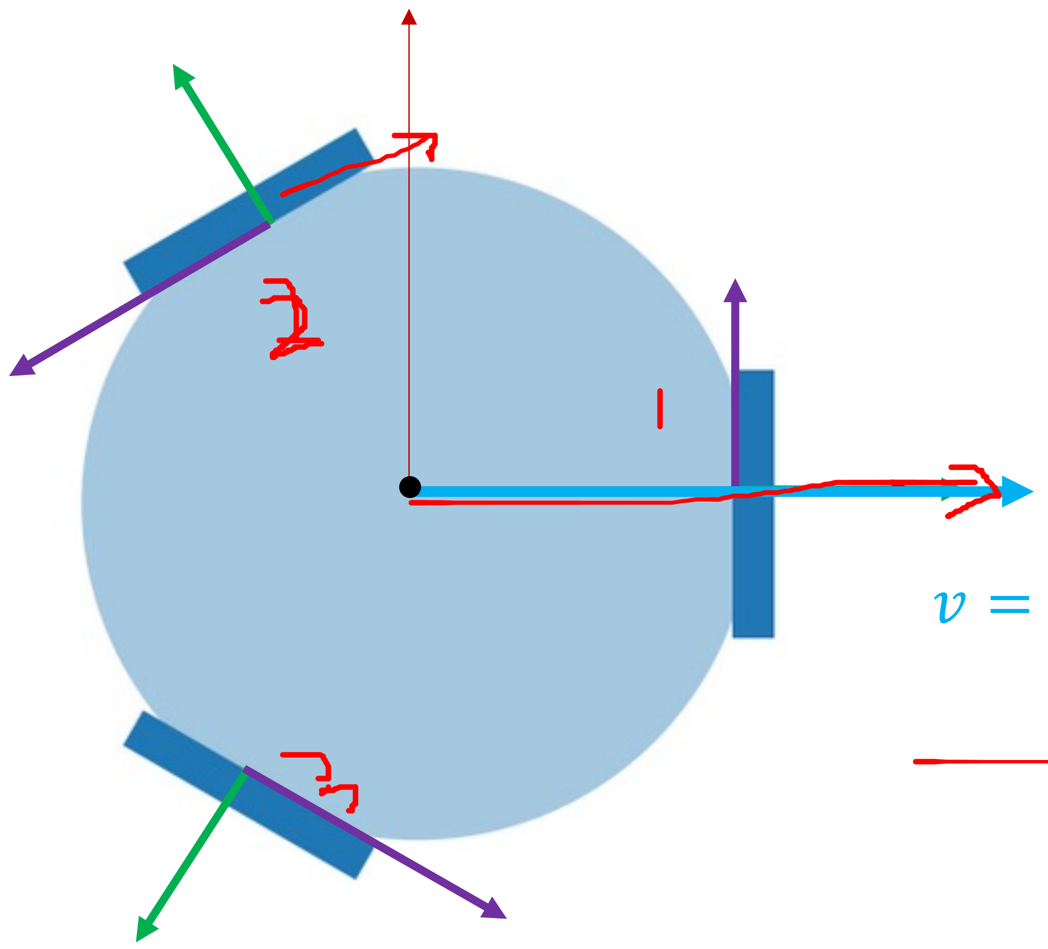
$$\begin{bmatrix} v_{\parallel}^1 \\ v_{\parallel}^2 \\ v_{\parallel}^3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.866 & -0.5 \\ 0.866 & -0.5 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

Example



$$\begin{bmatrix} v_{\parallel}^{1} \\ v_{\parallel}^{2} \\ v_{\parallel}^{3} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.866 & -0.5 \\ 0.866 & -0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ -0.5 \end{bmatrix}$$

Example



$$v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} v_{\parallel}^{1} \\ v_{\parallel}^{2} \\ v_{\parallel}^{3} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.866 & -0.5 \\ 0.866 & -0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.866 \\ 0.866 \end{bmatrix}$$

Wheel Jacobian

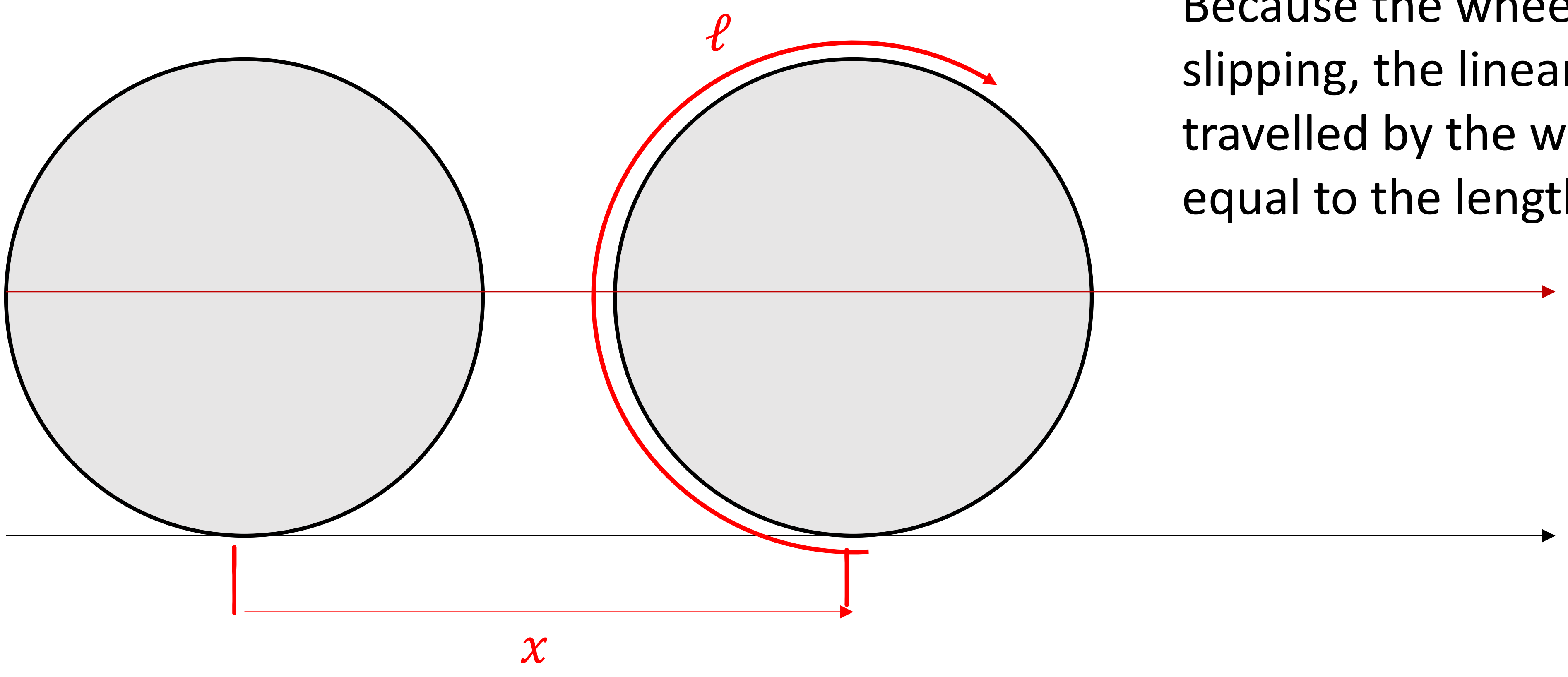
- A ***Jacobian*** matrix maps velocities in one coordinate system to velocities in another coordinate system.
- For our case, we want to map the velocity of the robot v to wheel rotation, specified as angular velocities ω^i for $i = 1,2,3$.
- The desired relationship is given by:

$$\begin{bmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \end{bmatrix} = J \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

- We'll need to relate rotation of the wheel to translation in the driving direction.

Rolling Without Slipping

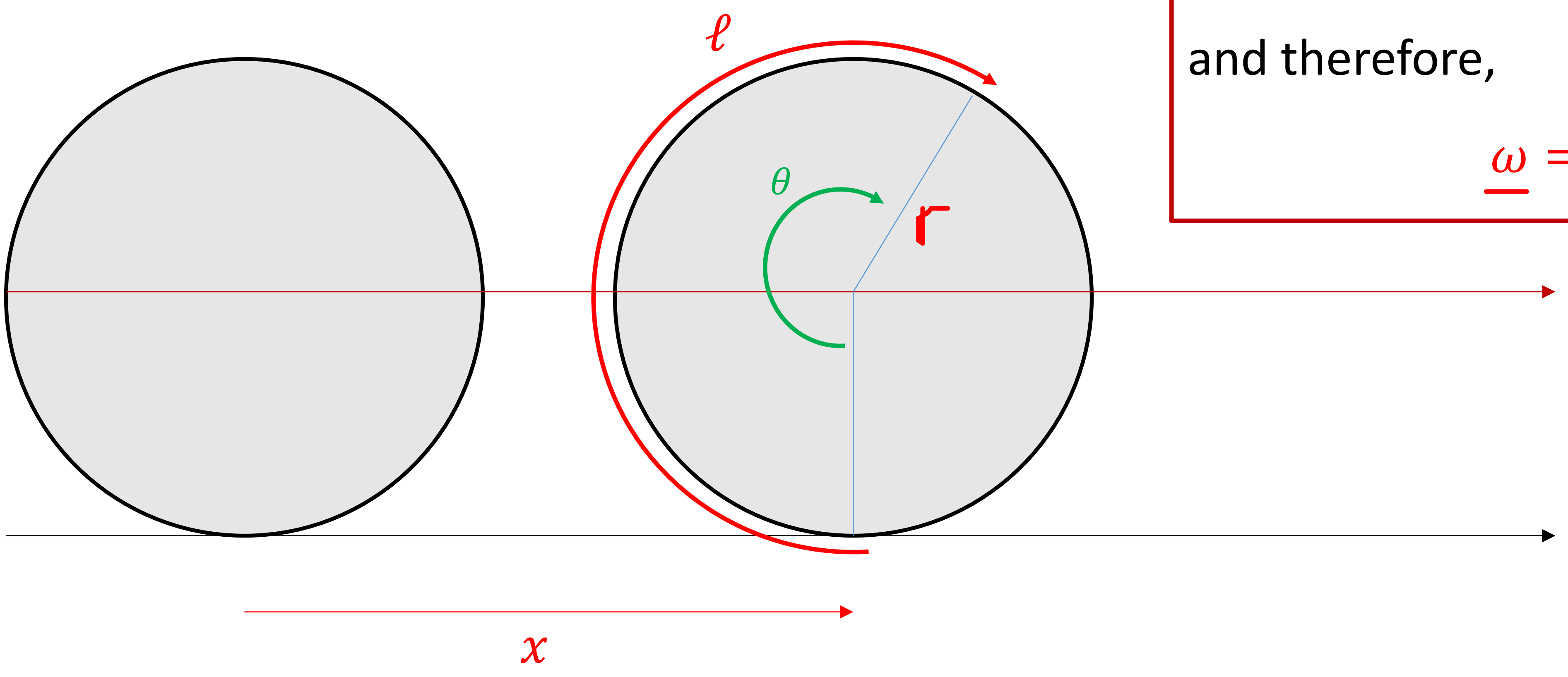
Suppose a wheel rolls without slipping a linear distance x.



Because the wheel rolls without slipping, the linear distance x travelled by the wheel center is equal to the length of the section ℓ.

Rolling Without Slipping

Using basic geometry, we know that $x = \ell = r\theta$.



Differentiating both sides, we obtain

$$\mathrm{v} = \frac{\mathrm{d}}{\mathrm{dt}}x = \frac{\mathrm{d}}{\mathrm{dt}}\ell = r\frac{d}{dt}\theta = r\underline{\omega}$$

and therefore,

$$\underline{\omega} = \frac{1}{r}v$$

Mapping Robot Velocity to Wheel Rotation

Combining these results, we obtain our final, Jacobian relationship:

$$v_{\parallel}^{i} = -v_x \sin\theta^{i} + v_y \cos\theta^{i}$$

$$v_{\perp}^{i} = v_x \cos\theta^{i} + v_y \sin\theta^{i}$$

$$\omega^{i} = \frac{1}{r} v_{\parallel}^{i}$$

$$\begin{bmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \end{bmatrix} = \frac{1}{r} \underbrace{\begin{bmatrix} -\sin\theta^1 & \cos\theta^1 \\ -\sin\theta^2 & \cos\theta^2 \\ -\sin\theta^3 & \cos\theta^3 \end{bmatrix}}_{\text{This is the Jacobian matrix, } J} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\begin{bmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 0 & 1 \\ -0.866 & -0.5 \\ 0.866 & -0.5 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

Discrete Time Motion Model

- The control input for our robot is a linear velocity v (which is converted to angular velocities for each wheel).

- We could model the motion of the robot using a differential equation: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

- It’s much simpler to use a discrete time model for the position of the robot:

$$\rightarrow \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_x \Delta T \\ y_t + v_y \Delta T \end{bmatrix} = \begin{bmatrix} x_t + u_x \\ y_t + u_y \end{bmatrix}$$

- If the motion of the robot happened to be deterministic and error-free, this would be all we need.
- Later, we’ll assume that the motion model is stochastic, and show how to model uncertainty using continuous probability density functions.

Limitations of our Model

The model we developed for omni-wheeled robots made several simplifications to what we might find in real applications:

- We conveniently aligned the robot’s coordinate system to a global world coordinate frame. Specifying the angle θ^i was simple, because it was specified in a coordinate frame that was fixed w.r.t. to the robot.
- Real robots sometimes rotate. We could accomplish this with the exact same robot by adding a rotational component to the robot velocity (i.e., robot angular velocity):

$$v = \begin{bmatrix} \omega^{robot} & v_x & v_y \end{bmatrix}^T$$

- If the robot rotates, then we’ll need to represent its orientation w.r.t. the global coordinate frame, since the steering directions of the wheels will change if the robot rotates.

Mecanum Wheels

- We can make the wheels a bit more interesting by changing the orientation of the “roller” wheels that allow sliding – ***Mecanum Wheels***.
- The math is (only) slightly more complex, but we won’t go further in this course.

Kinematics of Differential Drive Robots

Our logistics robot had super simple kinematics:

- Thanks to omni-wheels, the logistics robot could roll in any direction at any time.
- Because of this, there was no need to pay attention to the orientation of the robot.
- We didn’t really worry about a body-attached coordinate frame, since the robot frame was always parallel to the world frame.

Differential Drive Robots don’t have omni-wheels…

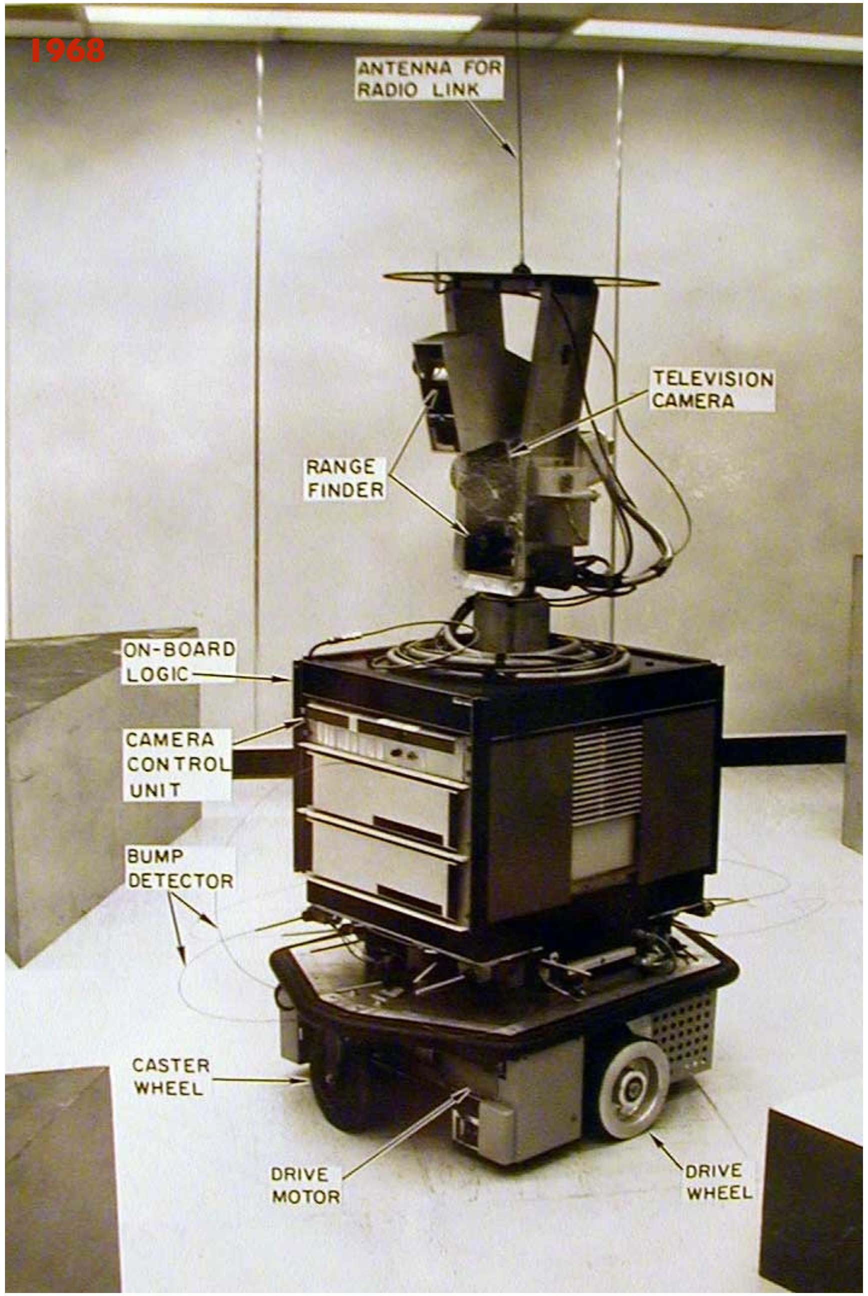
- The kinematics (relationship between input commands and robot motion) are more interesting.
- We need to explicitly pay attention to the orientation.

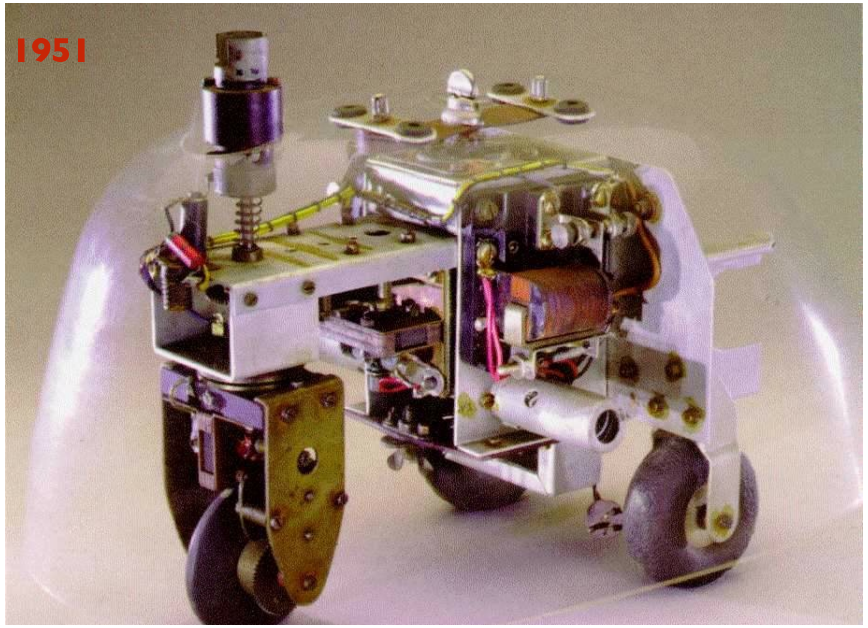
Mobile Robots

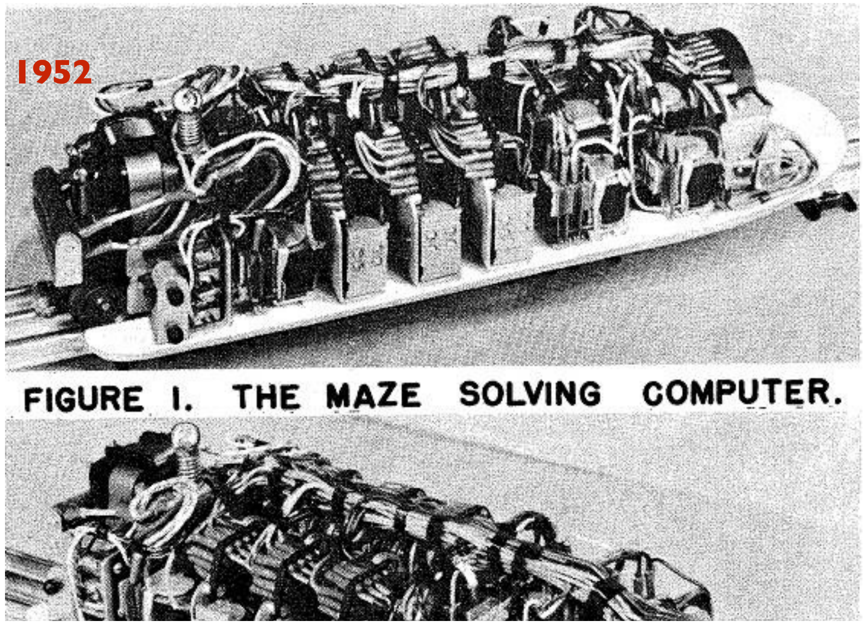
- There are many kinds of wheeled mobile robots.
- We have seen omni-directional robots.
- Now we'll study *differential drive robots.*

Mobile Robot Kinematics

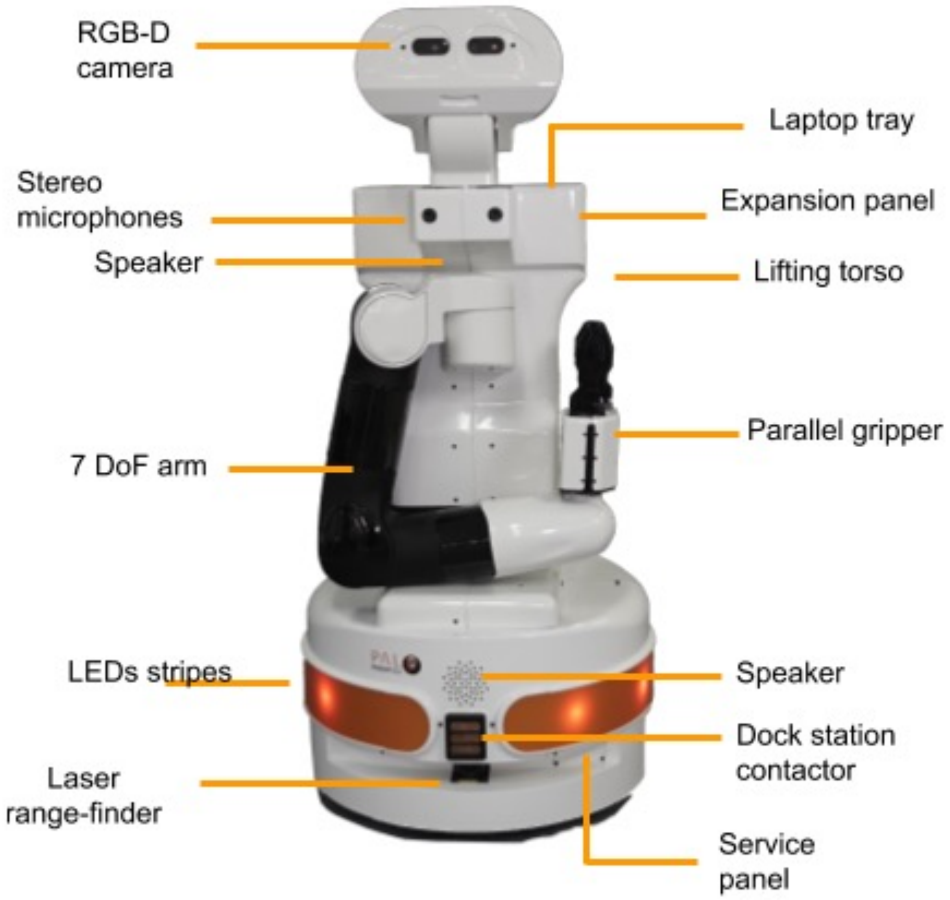
- Relationship between input commands (e.g., wheel velocity) and pose of the robot, not considering forces. *If the wheels turn at a certain rate, what is the resulting robot motion?*
- No direct way to measure pose (unless we sensorize the environment), but we can integrate velocity (odometry) to obtain a good estimate.

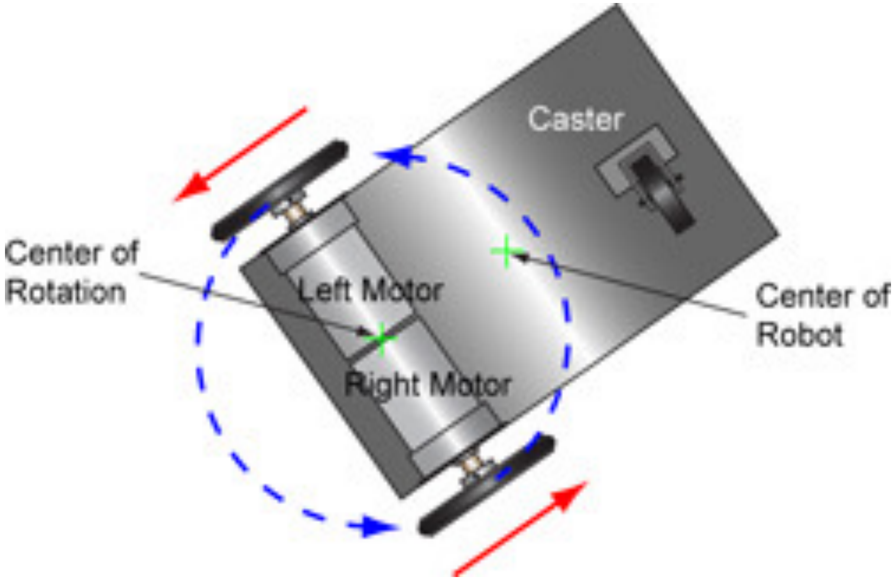

1968
ANTENNA FOR RADIO LINK
TELEVISION CAMERA
RANGE FINDER
ON-BOARD LOGIC
CAMERA CONTROL UNIT
BUMP DETECTOR
CASTER WHEEL
DRIVE MOTOR
DRIVE WHEEL

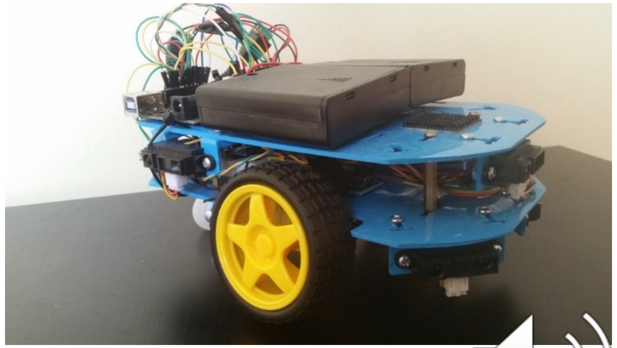

1951


1952
FIGURE I. THE MAZE SOLVING COMPUTER.

Differential Drive Robots

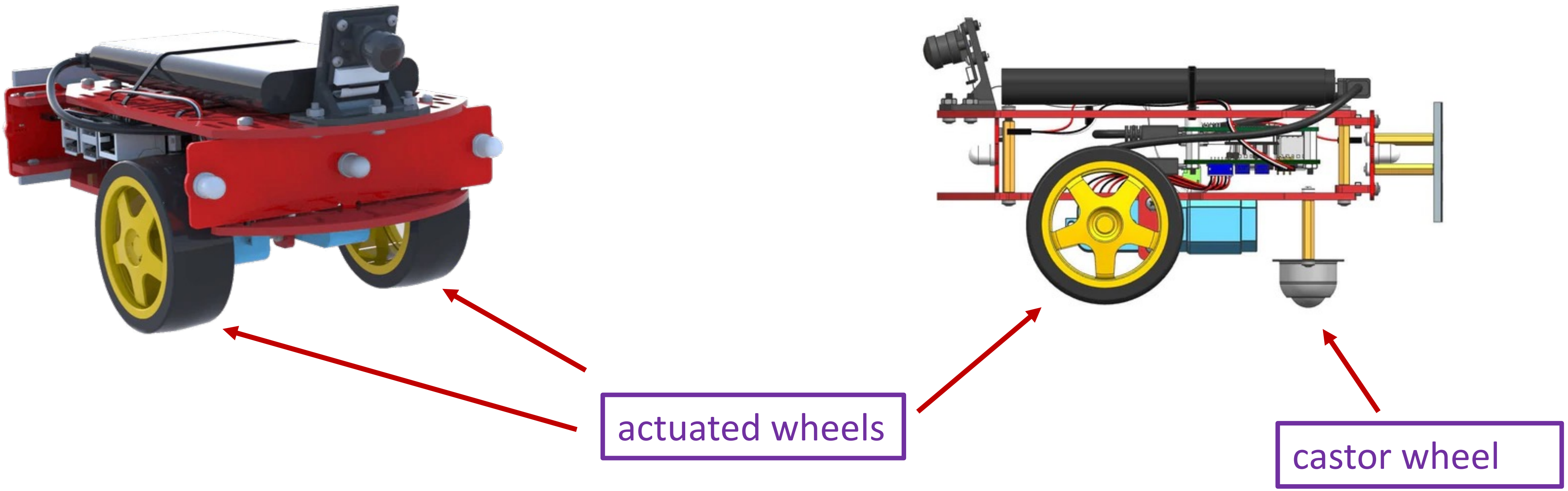






Two wheels with a common axis, and that can spin independently

The Duckiebot Platform

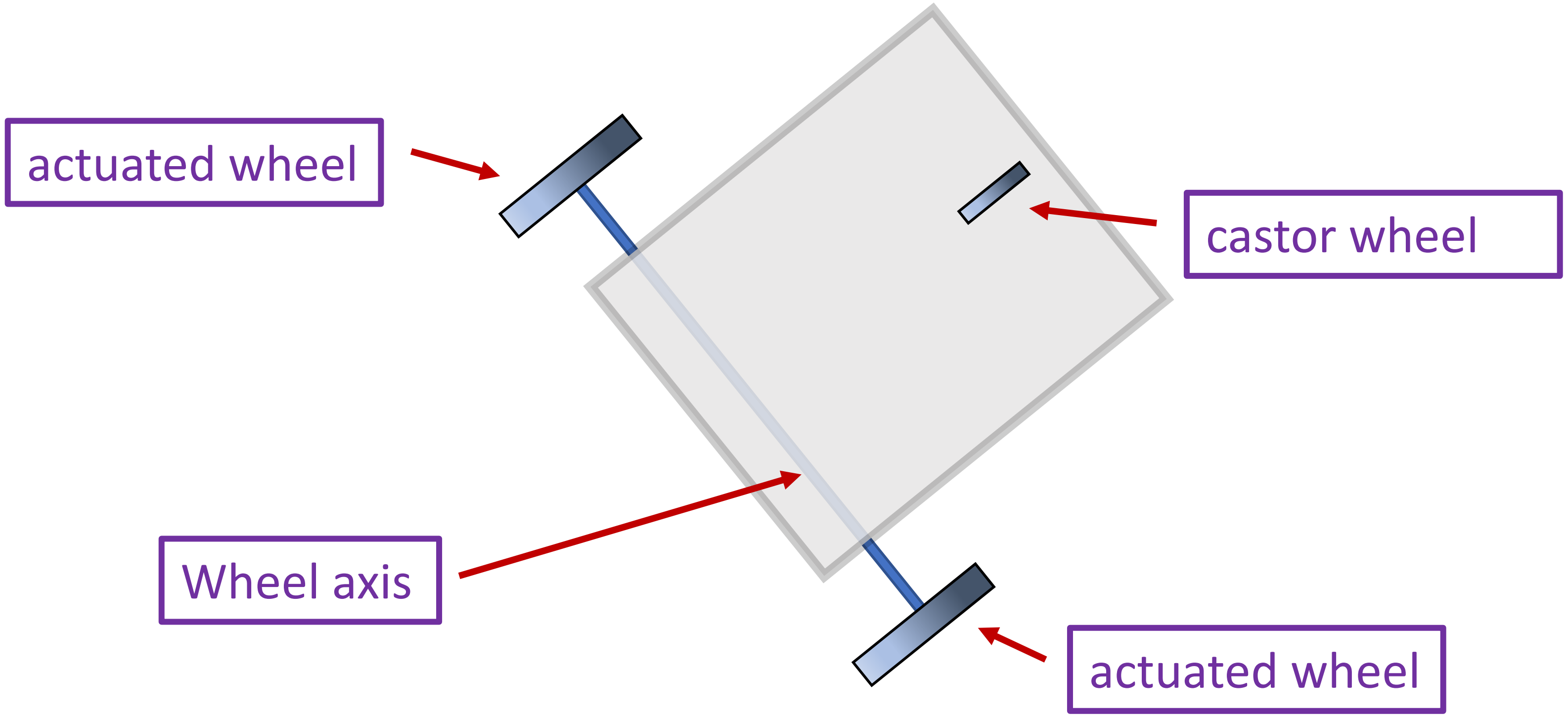


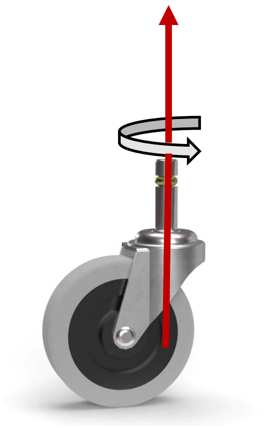
A typical DDR, with two actuated wheels in front, and a passive castor wheel in the back.

Differential Drive Robots

Differential drive robots (aka DDRs):

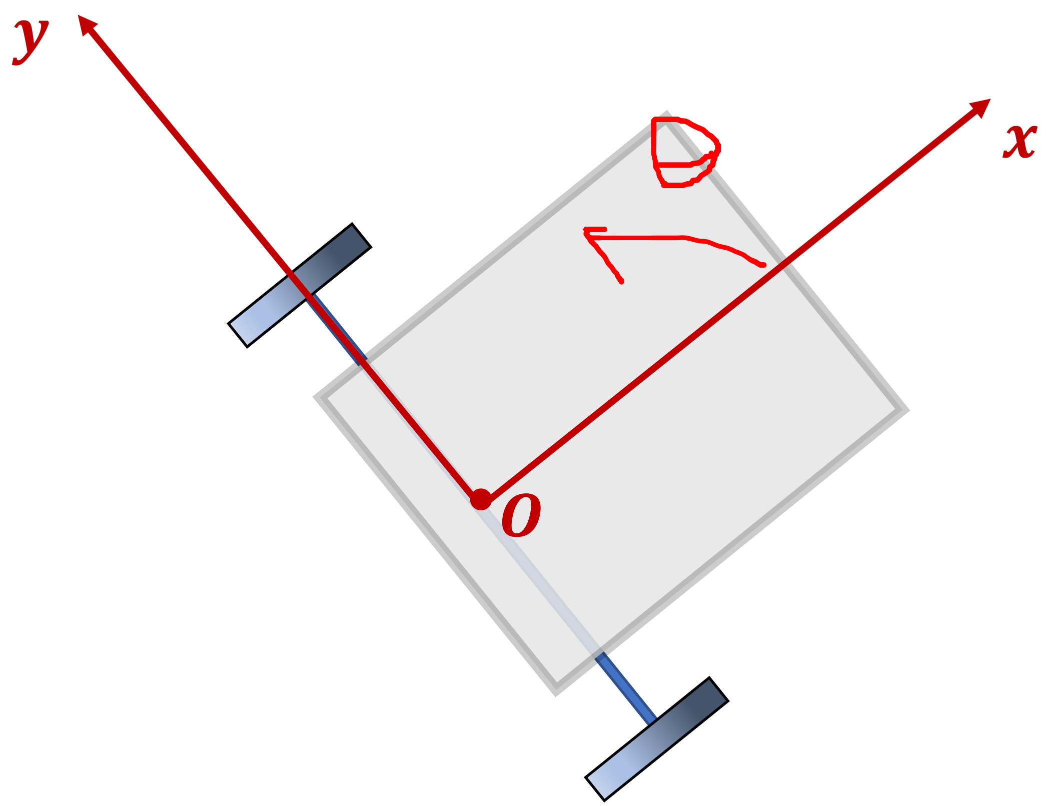
- Two actuated wheels that share an axis
- A castor wheel that rotates freely, mainly to stabilize the robot (three points define a plane – castor wheel keeps the robot from tipping over).





A castor wheel is able to spin freely about the vertical axis.

Differential Drive Robots



To specify the position and orientation of the DDR, we attach a coordinate frame to the robot.

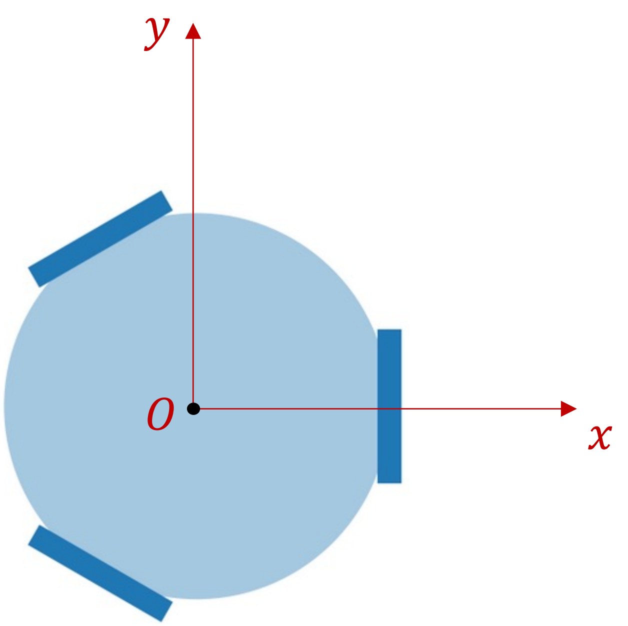
- This frame is called the ***body-attached frame***, or the robot frame.
- The body attached frame is ***rigidly*** attached to the robot: it translates and rotates with the robot.
- The origin of the body-attached frame is located at the midpoint between the two wheels along their axis of rotation.
- The x-axis is the steering (or driving) direction of the robot.
- The y -axis is coincident with the common axis.

Configuration Space

- A ***configuration*** is a complete specification of the position of every point in a robot system.
- The ***configuration space*** is the set of all configurations.
- We use q to denote a point in a configuration space Q.

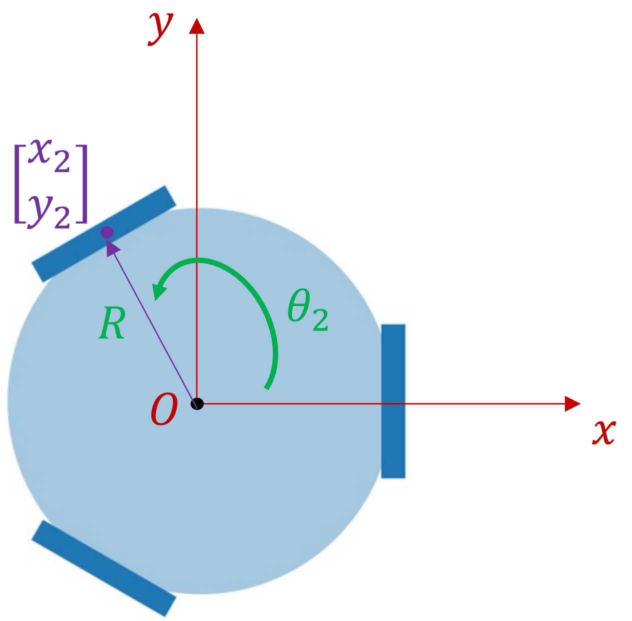
Example:

- Our logistics robot was able to translate in the plane.
- It's orientation never changed (i.e., it could not rotate).
- We can attach a coordinate frame with origin at the center of the robot, and axes parallel to the world x- and y-axes.
- If we know the x, y coordinates of this coordinate frame, we can easily determine the location of any desired point on the robot w.r.t. the world coordinate frame (in the x- y plane).



Configuration Space

- A ***configuration*** is a complete specification of the position of every point in a robot system.
- The ***configuration space*** is the set of all configurations.
- We use q to denote a point in a configuration space Q.



Example:

- Our logistics robot was able to translate in the plane.
- It's orientation never changed (i.e., it could not rotate).
- We can attach a coordinate frame with origin at the center of the robot, and axes parallel to the world x- and y-axes.
- If we know the x, y coordinates of this coordinate frame, we can easily determine the location of any desired point on the robot w.r.t. the world coordinate frame (in the x- y plane).
- For example, if the robot has radius R, then the center of wheel 2 is:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_{robot} + R \cos \theta_2 \\ y_{robot} + R \sin \theta_2 \end{bmatrix}$$

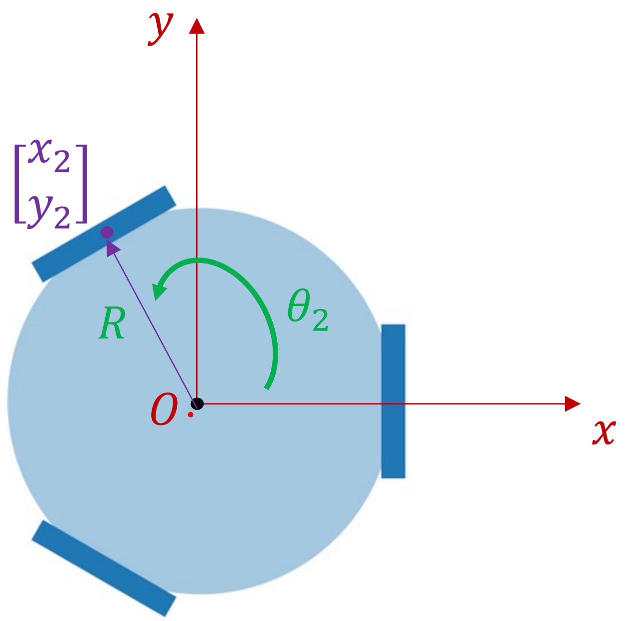
Configuration Space

- A ***configuration*** is a complete specification of the position of every point in a robot system.
- The ***configuration space*** is the set of all configurations.
- We use q to denote a point in a configuration space Q.

In this example, the configuration space is easy to characterize:

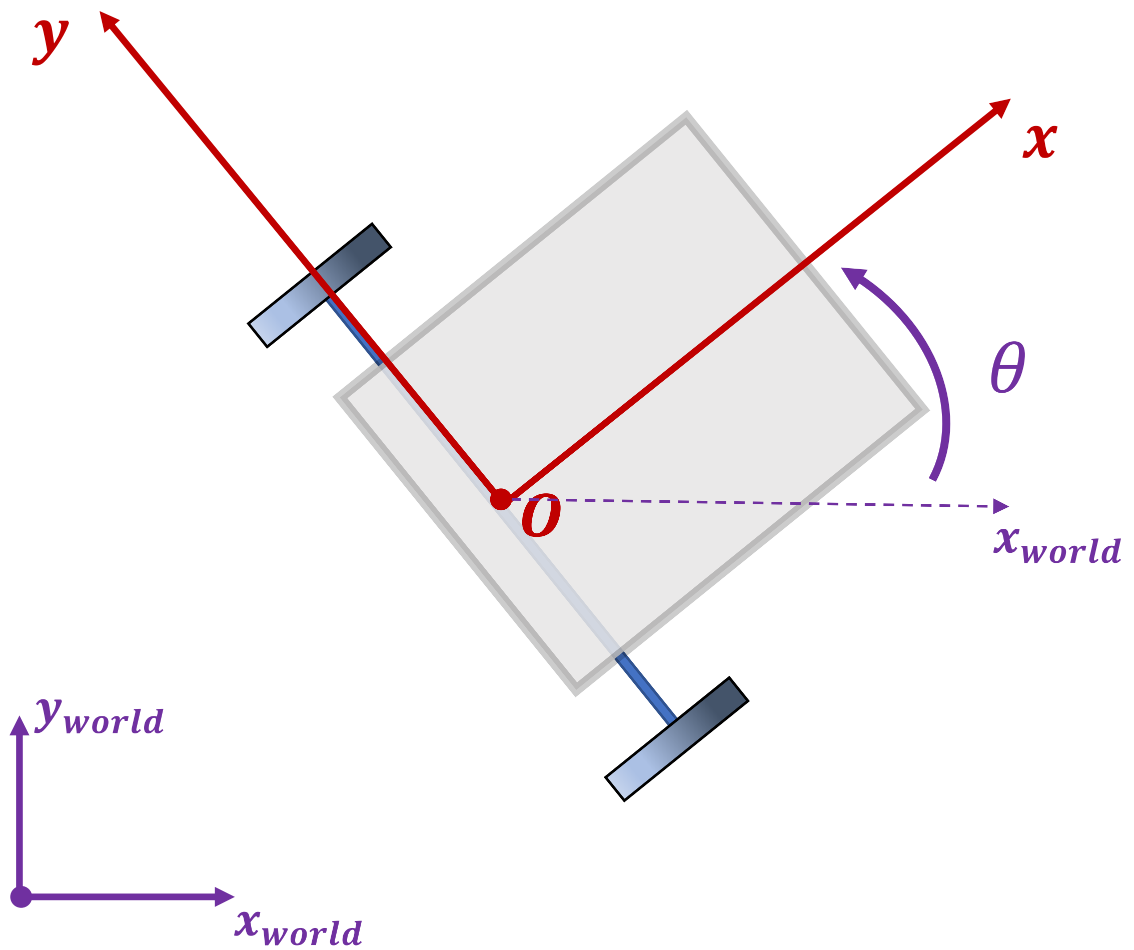
$$Q = \mathfrak{D} \subset \mathbb{R}^2$$
$$q = (x, y) \in \mathfrak{D}$$

where $\mathfrak{D}$ is the floor space of the warehouse.



- Given $q = (x, y)$, we can calculate the position of ***any*** point on the robot.
- Note: This assumes, of course, that we have a model of our robot, which we do.

Configuration Space for a DDR



Because our DDR can rotate in the plane, it is necessary to know both the position and the orientation of the body-attached frame to specify a configuration:

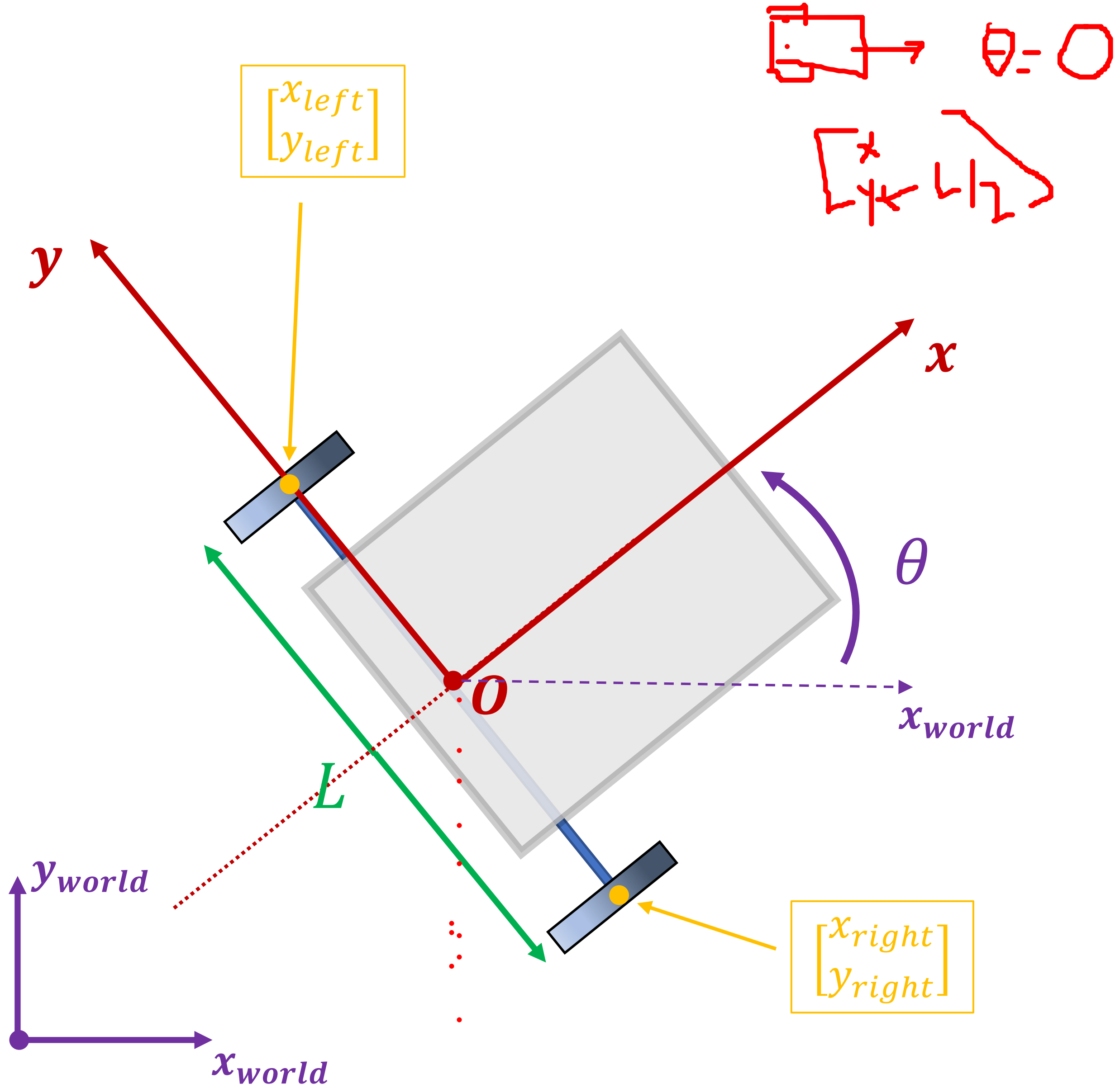
$$Q = \mathbb{R}^2 \times [0,2\pi)$$

$$q = (x, y, \theta) \in Q$$

If we know the configuration, $q = (x, y, \theta)$, we can compute the location of any point on the robot.

Let's start with the wheel centers.

Configuration Space for a DDR



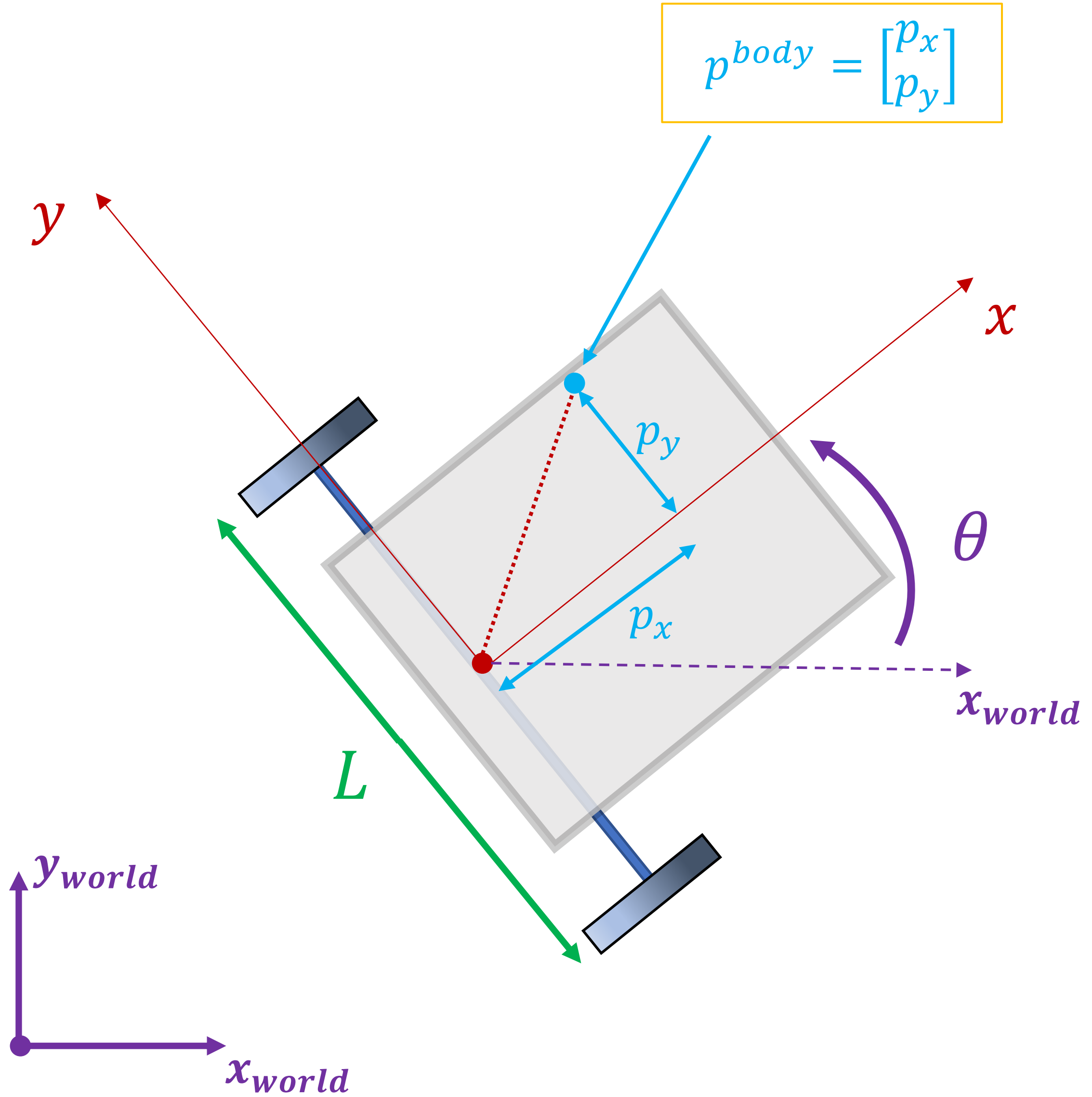
If the robot is in configuration $q = (x, y, \theta)$, the left and right wheel centers are located at:

$$\begin{bmatrix} x_{left} \\ y_{left} \end{bmatrix} = \begin{bmatrix} x - \frac{L}{2}\sin\theta \\ y + \frac{L}{2}\cos\theta \end{bmatrix}$$

and

$$\begin{bmatrix} x_{right} \\ y_{right} \end{bmatrix} = \begin{bmatrix} x + \frac{L}{2}\sin\theta \\ y - \frac{L}{2}\cos\theta \end{bmatrix}$$

Kinematics of DDRs



- We can generalize this to any point p on the DDR.
- Suppose the coordinates of p in the body frame are given by

$$p^{body} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

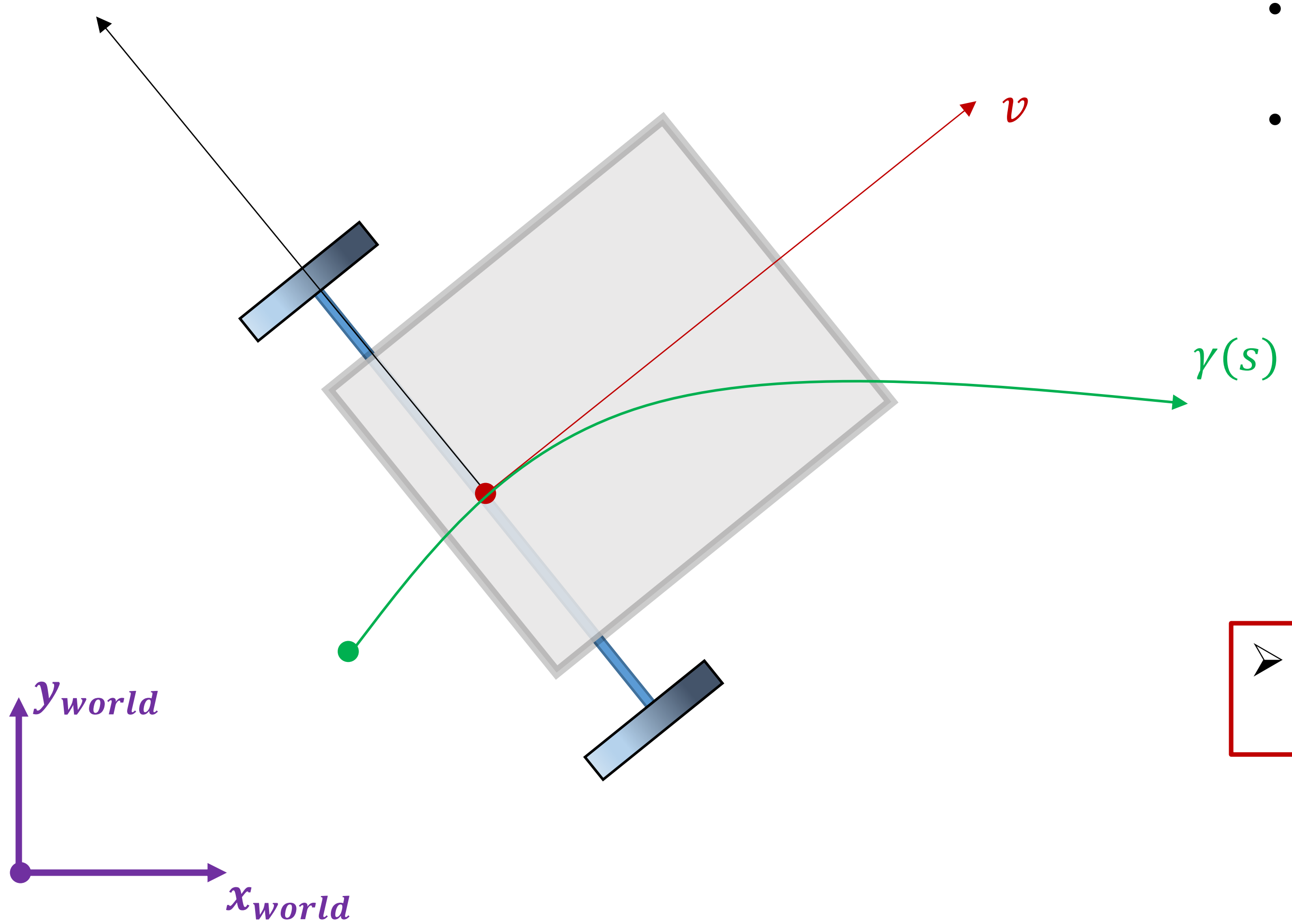
- If the robot is in configuration $q = (x, y, \theta)$, the coordinates of p in the world frame are given by:

$$p^{world} = \begin{bmatrix} x + p_x \cos\theta - p_y \sin\theta \\ y + p_x \sin\theta + p_y \cos\theta \end{bmatrix}$$

$$= R_{body}^{World} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

- ***This is just a homogeneous transformation!***
- ***Note: The Homogeneous transformation is parameterized by the configuration $q = (x, y, \theta)$.***

Linear Velocity of the DDR

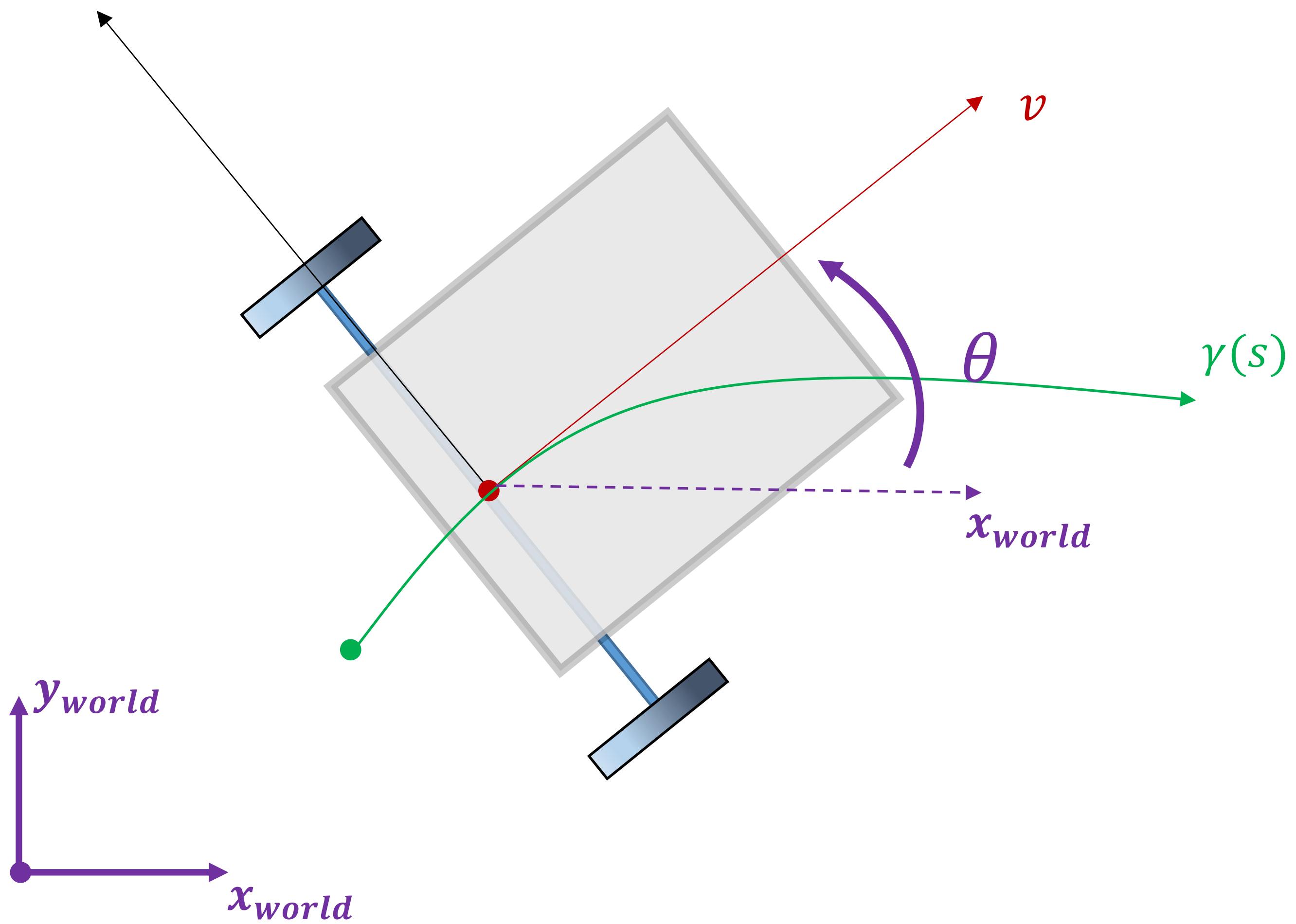


Differential Drive Robots are very different from robots with omni-wheels:

- The wheels roll without slipping – *no sideways motion*.
- The instantaneous velocity of the robot is always in the steering direction.
- The velocity perpendicular to the steering direction is always zero.

> ➢ If the robot follows the curve $\gamma(s)$, the instantaneous velocity v is tangent to γ.

Velocity of the DDR



- Since the robot cannot move in the direction of the body-attached y-axis, its linear velocity, when expressed with respect to the body frame is:

$$v^{body,linear} = \begin{bmatrix} v_x \\ 0 \end{bmatrix} \quad v_y = 0$$

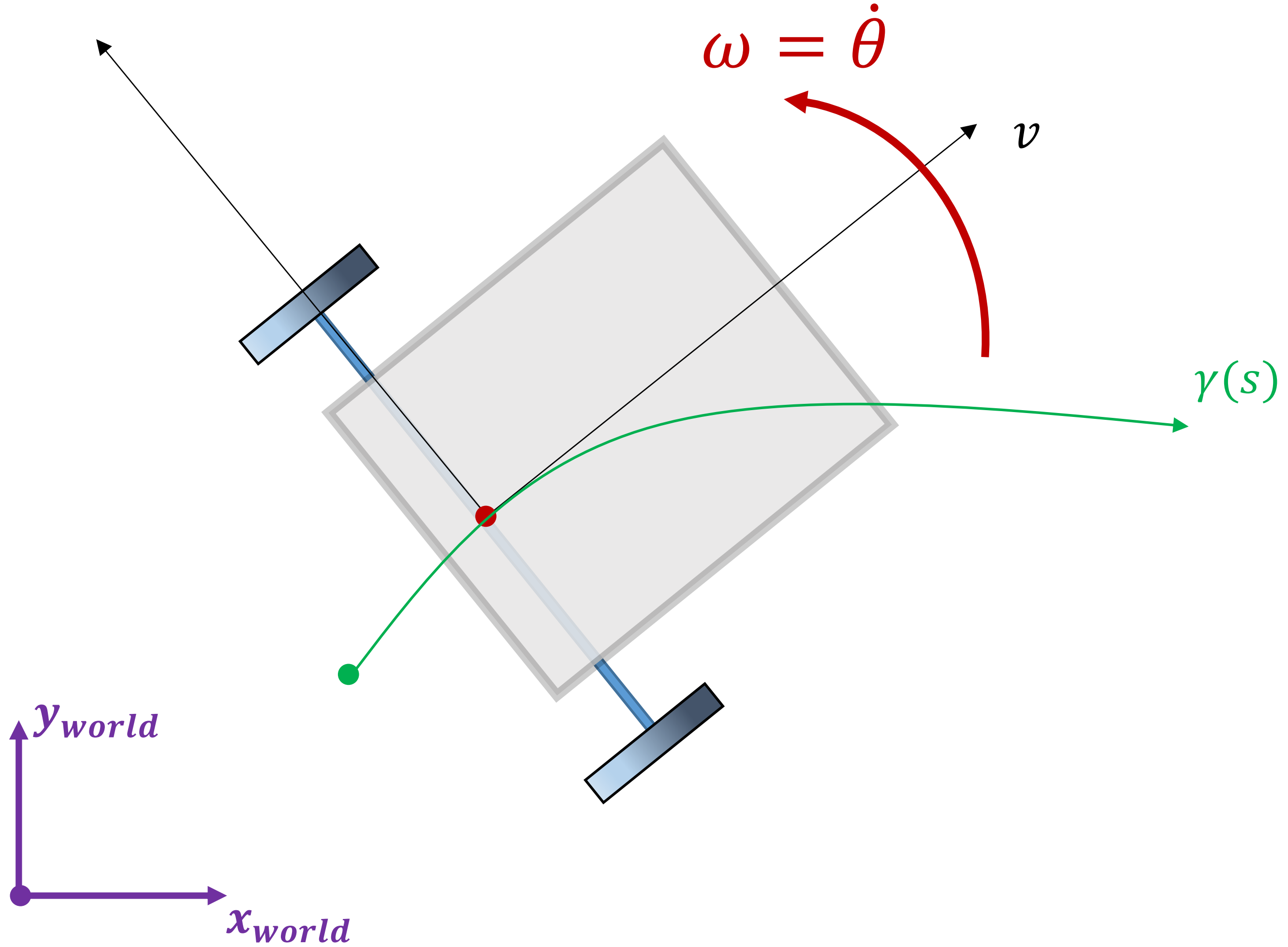
- The steering direction, expressed w.r.t. the world frame, is given by:

$$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

- Therefore, when the robot is in configuration $q(s) = (x, y, \theta)$, its linear velocity is expressed with respect to the world frame by:

$$v^{world,linear} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \end{bmatrix}$$

Angular Velocity of the DDR



- The orientation of the robot is given by the angle θ.
- Since this robot is able to rotate, θ can be considered as a function of time.
- We define the robot's angular velocity as

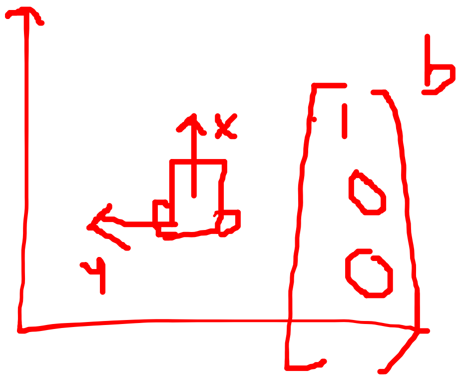
$$\omega = \frac{d}{dt}\theta = \dot{\theta}$$

- Note that the positive sense for ω is defined using the right-hand rule: point the thumb of your right hand in the direction of the world z-axis, and your fingers will curl in the positive θ direction.

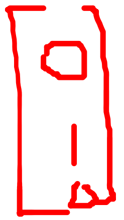
Total Velocity of the DDR

- The velocity of the DDR includes both the linear and angular velocities.
- We stack these into a single vector to describe the robot's instantaneous velocity w.r.t. the body frame of the world frame:

$$v^{body} = \begin{bmatrix} v_x \\ 0 \\ \dot{\theta} \end{bmatrix}, \qquad v^{world} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \dot{\theta} \end{bmatrix}$$

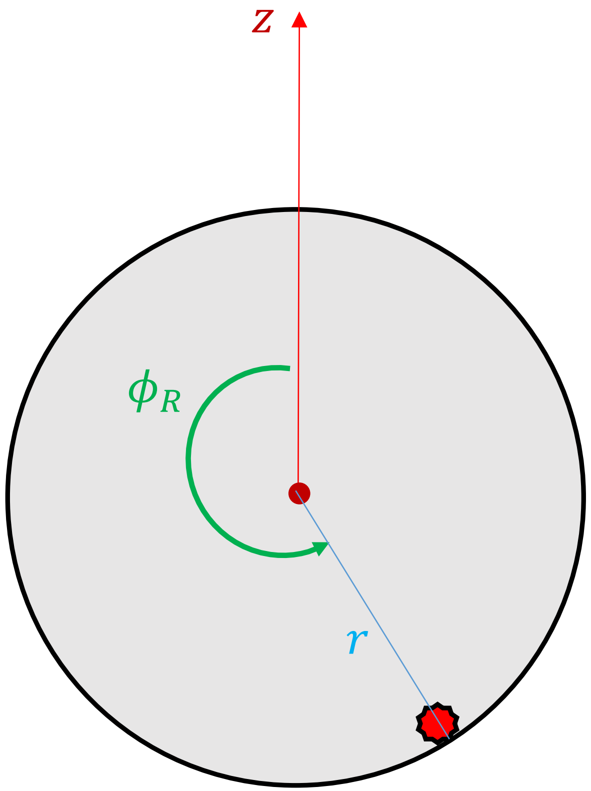


➢ Note that the z-axis of the body-attached frame ***<u>is the same as</u>*** the z-axis of the world frame, so that the angular velocity is given by $\dot{\theta}$ for both of these coordinate frames.



Wheel Actuation and DDR Velocity

- The two wheels of the DDR are independently actuated, and able to spin in both directions.
- Let ϕ_R and ϕ_L denote the instantaneous orientation of the right and left wheels (e.g., the angle from the world z-axis to some identifiable mark on the wheel.
- The angular speeds of the wheels are therefore given by $\dot{\phi}_R$ and $\dot{\phi}_L$.



As we saw with the omni-directional robot, the relationship between forward speed of the wheel and its angular speed is given by

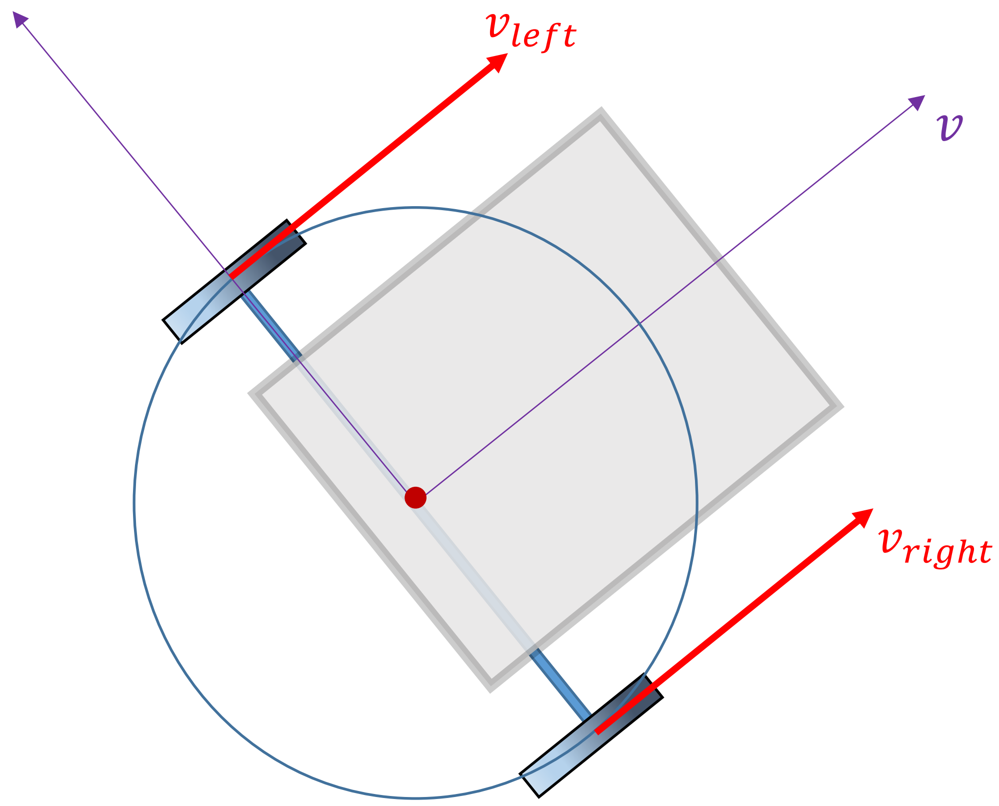
$$\frac{\mathrm{d}}{\mathrm{dt}}x = r\dot{\phi}$$

and therefore, since the wheel rolls without slipping, and $v_y = 0$, we have

$$\dot{\phi} = \frac{v_x}{r}$$

Wheel Actuation and DDR Velocity

When the two wheels turn with the same angular speed, $\dot{\phi}_R = \dot{\phi}_L$, the robot moves with pure translation.



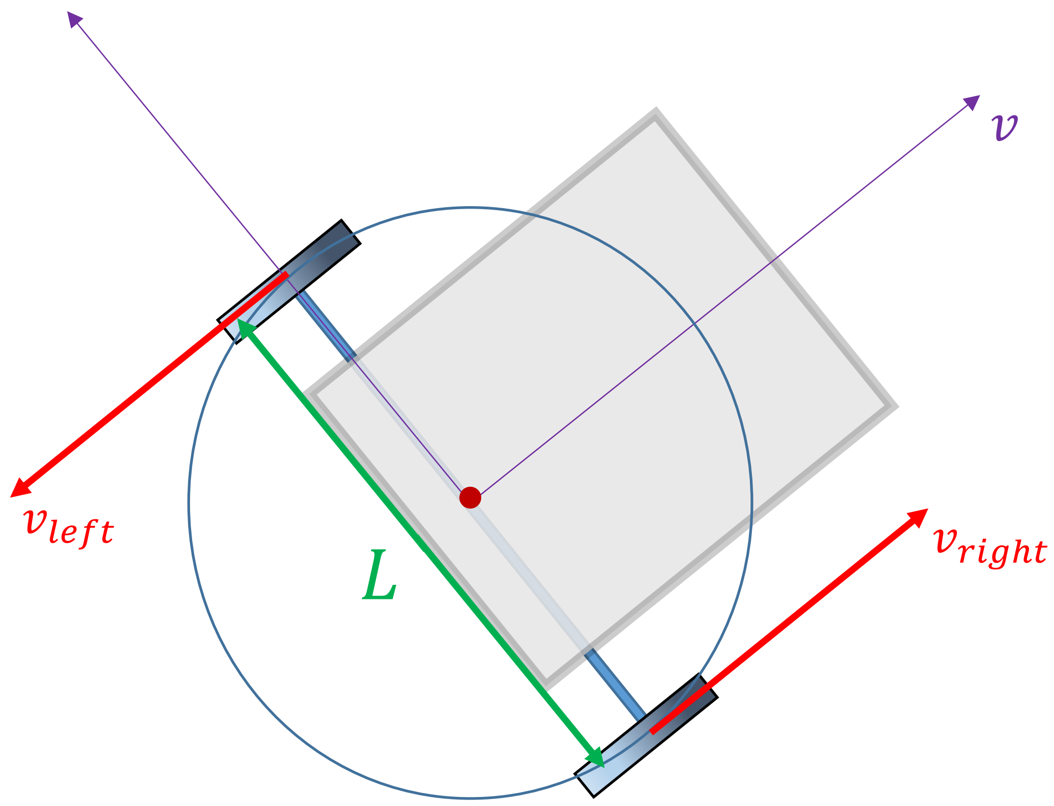
In this case,

- $v_{left} = v_{right}$
- Both v_{left} and v_{right} are parallel to the steering direction.
- The robot's angular velocity is zero (i.e., $\omega = 0$).
- The angular wheel speed is related to the robot's linear velocity by

$$\dot{\phi}_R = \dot{\phi}_L = \frac{v_x}{r}$$

Total DDR Velocity

When the two wheels turn with the opposite angular velocity, $\dot{\phi}_R = -\dot{\phi}_L$, the robot moves with pure rotation.



In this case,

- $v_{left} = -r\dot{\phi}_L$ and $v_{right} = r\dot{\phi}_R$
- Both v_{left} and v_{right} are tangent to the circle centered at the origin of the body-attached frame w/radius $0.5L$.
- The robot's angular velocity satisfies *the eqn for circular motion:*

$$\frac{L}{2}\omega = v_{right} = r\dot{\phi}_R$$

and

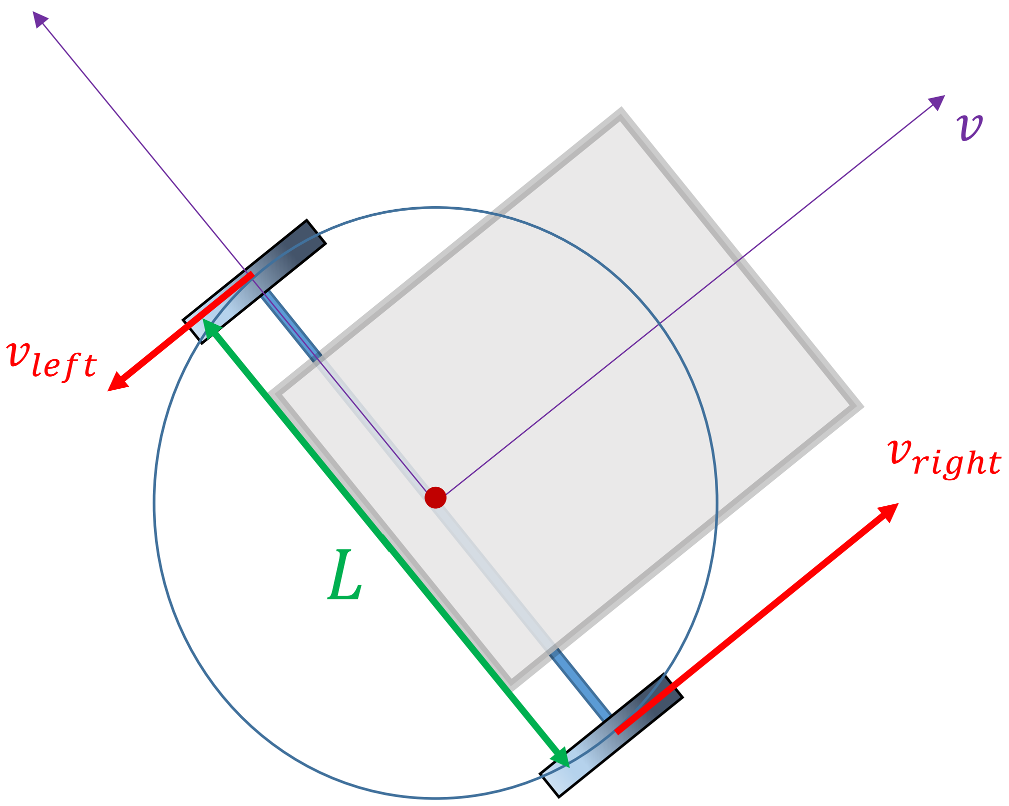
$$\frac{L}{2}\omega = -v_{left} = -r\dot{\phi}_L$$

- Rearranging terms, we obtain:

$$\dot{\phi}_R = \frac{L}{2}\frac{\omega}{r} \quad \text{and} \quad \dot{\phi}_L = -\frac{L}{2}\frac{\omega}{r}$$

Wheel Actuation and DDR Velocity

All other velocities are linear combinations of these two cases, and therefore we can apply superposition.



- For pure translation we have:

$$\dot{\phi}_R = \frac{v_x}{r} \quad \text{and} \quad \dot{\phi}_L = \frac{v_x}{r}$$

- For pure rotation we have:

$$\dot{\phi}_R = \frac{L}{2}\frac{\omega}{r} \quad \text{and} \quad \dot{\phi}_L = -\frac{L}{2}\frac{\omega}{r}$$

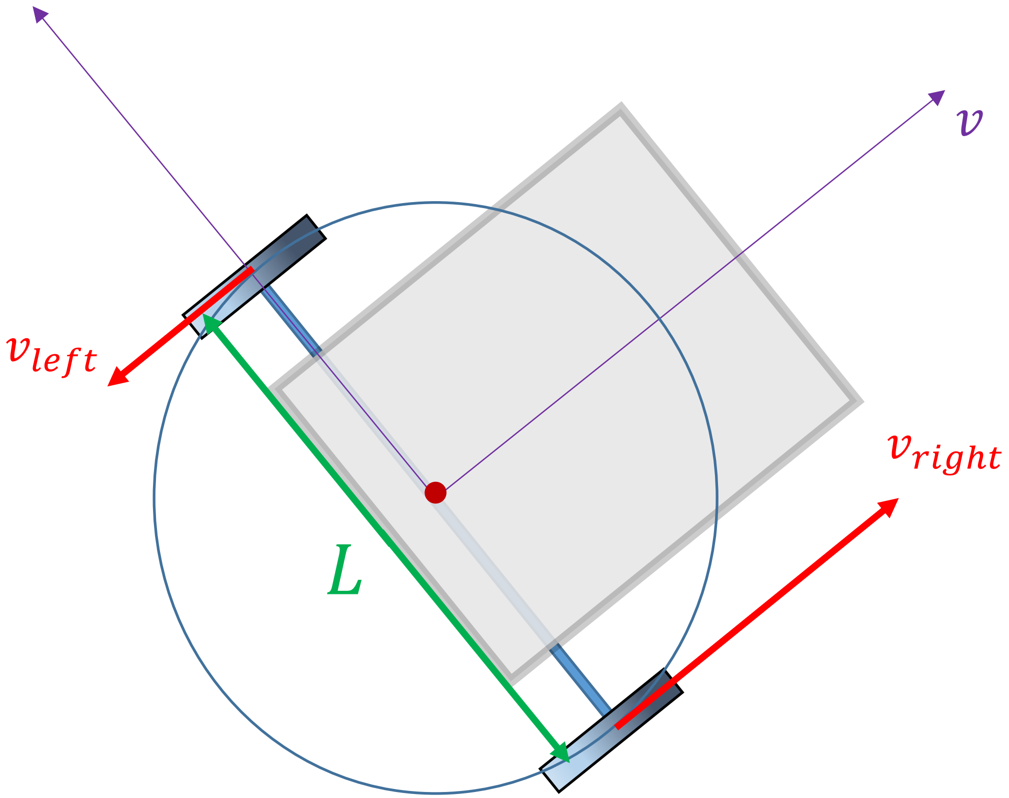
- Combining (adding) the two equations for $\dot{\phi}_R$ and $\dot{\phi}_L$ we obtain:

$$\dot{\phi}_R = \frac{L}{2}\frac{\omega}{r} + \frac{v_x}{r} \quad \text{and} \quad \dot{\phi}_L = -\frac{L}{2}\frac{\omega}{r} + \frac{v_x}{r}$$

➢ ***These two equations tell us how to choose $\dot{\phi}_R$ and $\dot{\phi}_L$ to achieve a desired velocity v, ω.***

Wheel Actuation and DDR Velocity

All other velocities are linear combinations of these two cases, and therefore we can apply superposition.



- The equations

$$\dot{\phi}_R = \frac{L}{2}\frac{\omega}{r} + \frac{v_x}{r} \quad \text{and} \quad \dot{\phi}_L = -\frac{L}{2}\frac{\omega}{r} + \frac{v_x}{r}$$

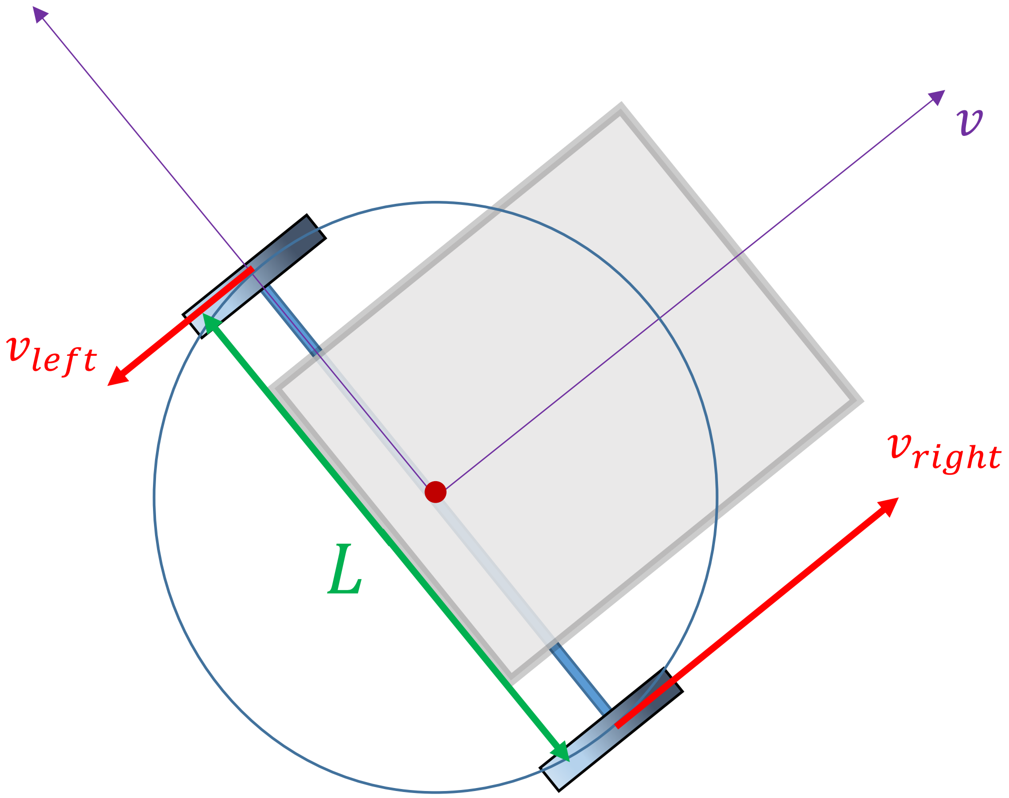
  are sometimes called *inverse* equations, since they solve the inverse problem: "what wheel speed (*input*) to I need to achieve a desired robot behavior (*output*)?"

- We can easily compute the forward mapping, from $\boldsymbol{\dot{\phi}_R}$ and $\boldsymbol{\dot{\phi}_L}$ to $\boldsymbol{v}, \boldsymbol{\omega}$ using simple algebra:

$$\dot{\phi}_R + \dot{\phi}_L = 2\frac{v_x}{r} \Rightarrow v_x = \frac{r}{2}\left(\dot{\phi}_R + \dot{\phi}_L\right)$$

$$\dot{\phi}_R - \dot{\phi}_L = \frac{L\omega}{r} \Rightarrow \omega = \frac{r}{L}\left(\dot{\phi}_R - \dot{\phi}_L\right)$$

Wheel Actuation and DDR Velocity



We can express these equations relative to the body-attached frame or the world frame:

$$v^{body} = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L) \\ 0 \\ \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L) \end{bmatrix}$$

and

$$v^{world} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L)\cos\theta \\ \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L)\sin\theta \\ \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L) \end{bmatrix}$$