Lablet 102: Wheeled robots: Motion Planning and Tracking

ARTPARK
AI & Robotics Technology Park

Lecture 1

Course instructor: Naveen Arulselvan

ARTPARK
AI & Robotics Technology Park

Labs based on Python notebooks (7)

ARTPARK
AI & Robotics Technology Park

- Simulating Unicycle/ Bicycle model

- Planning
  - A*/ Djikstra
    - Intra-city planning (using Open street maps and OSNX)
  - RRT*
  - Smooth polynomials for parking

- Tracking
  - PID / Pure-pursuit controller
  - Model Predictive control

- Avoidance
  - State lattice planner/ Trajectory tracking in Frenet-frame
  - Dynamic window avoidance
-

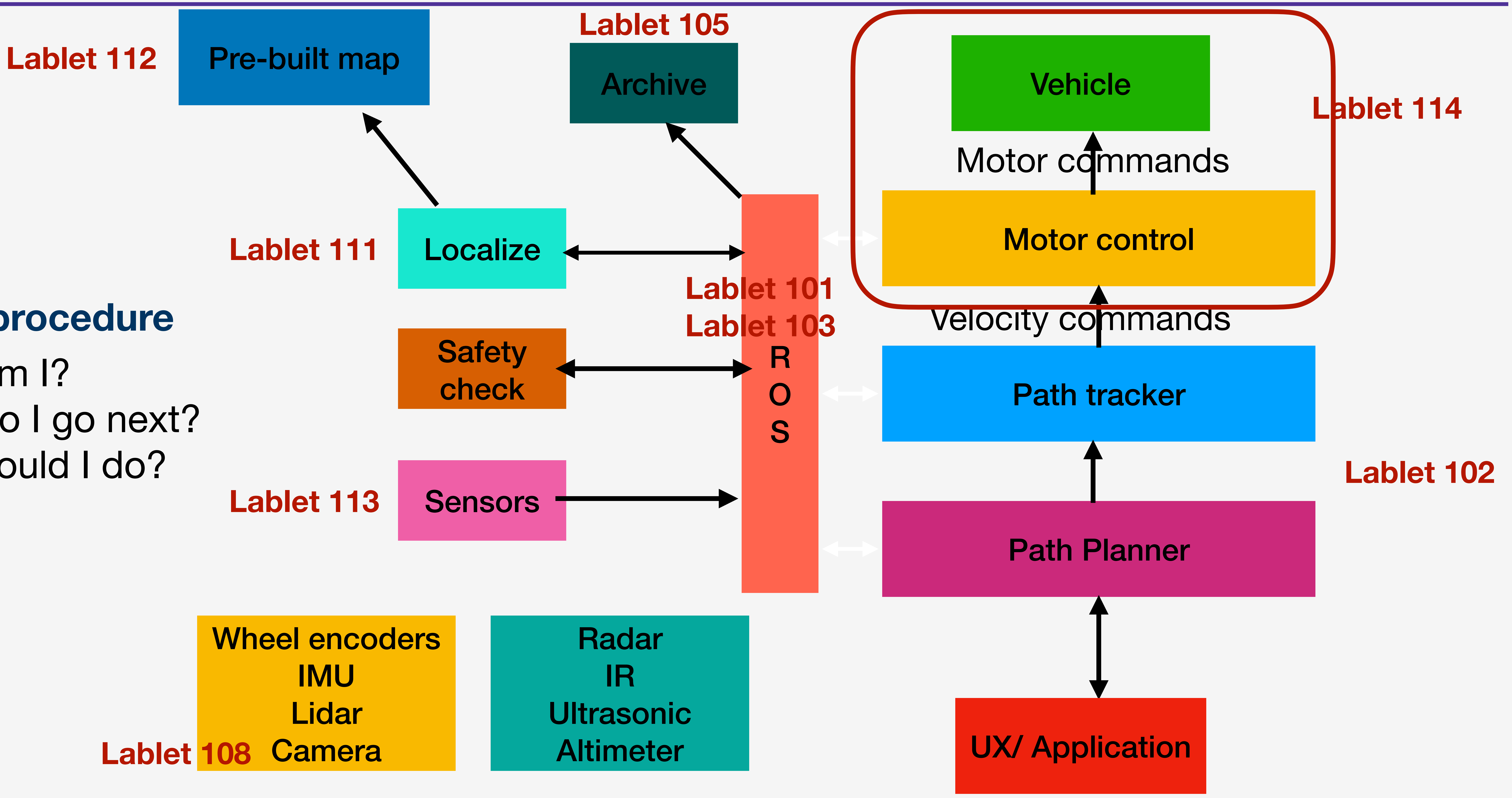
Course Takeaways

1. How to model robot motion?
2. How to plan paths for robot motion? (Planning)
3. How to ensure robot stays on the path? (Tracking)
4. How to avoid obstacles on the path? (Collision checking/ avoidance)

ARTPARK
AI & Robotics Technology Park

Autonomy and Control: Pub-Sub architecture

ART PARK
AI & Robotics Technology Park



Iterative procedure

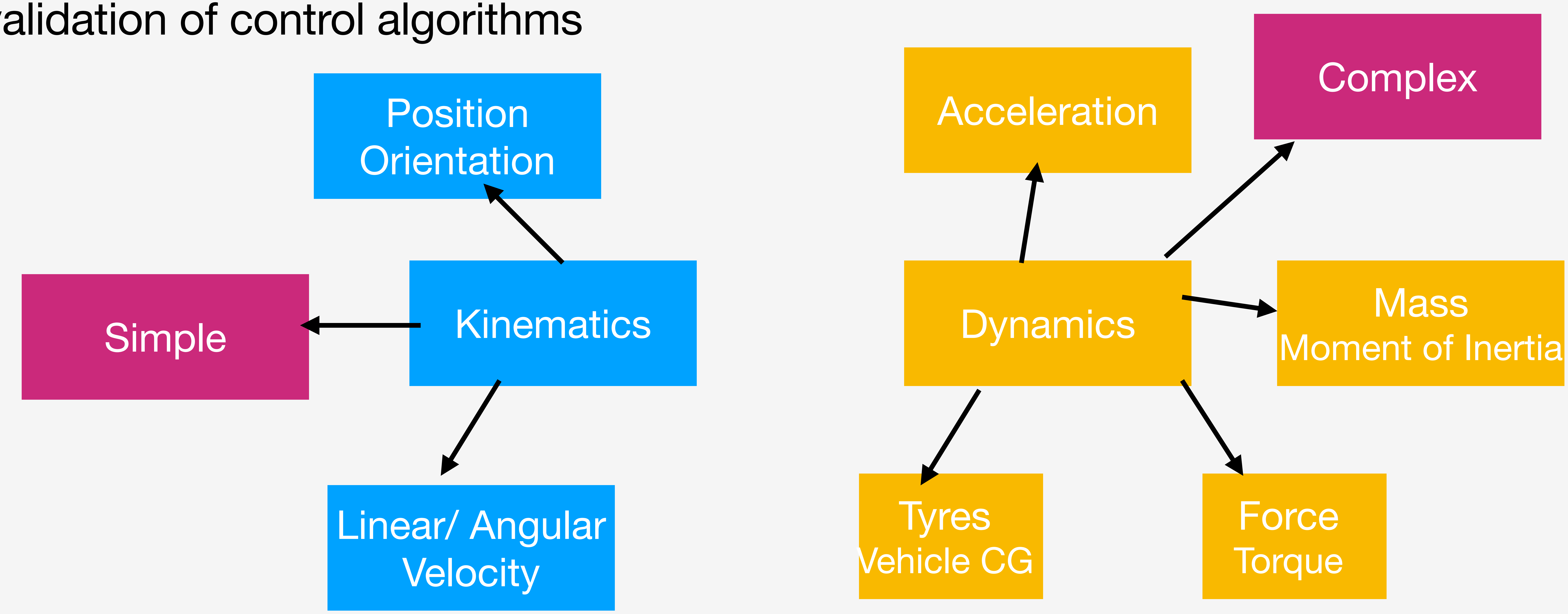
1. Where am I?
2. Where do I go next?
3. What should I do?

Lablet sequence subject to change

Kinematics and Dynamics

ART PARK
AI & Robotics Technology Park

Vehicle models needed for simulation/ validation of control algorithms



Simulations = cheap to test

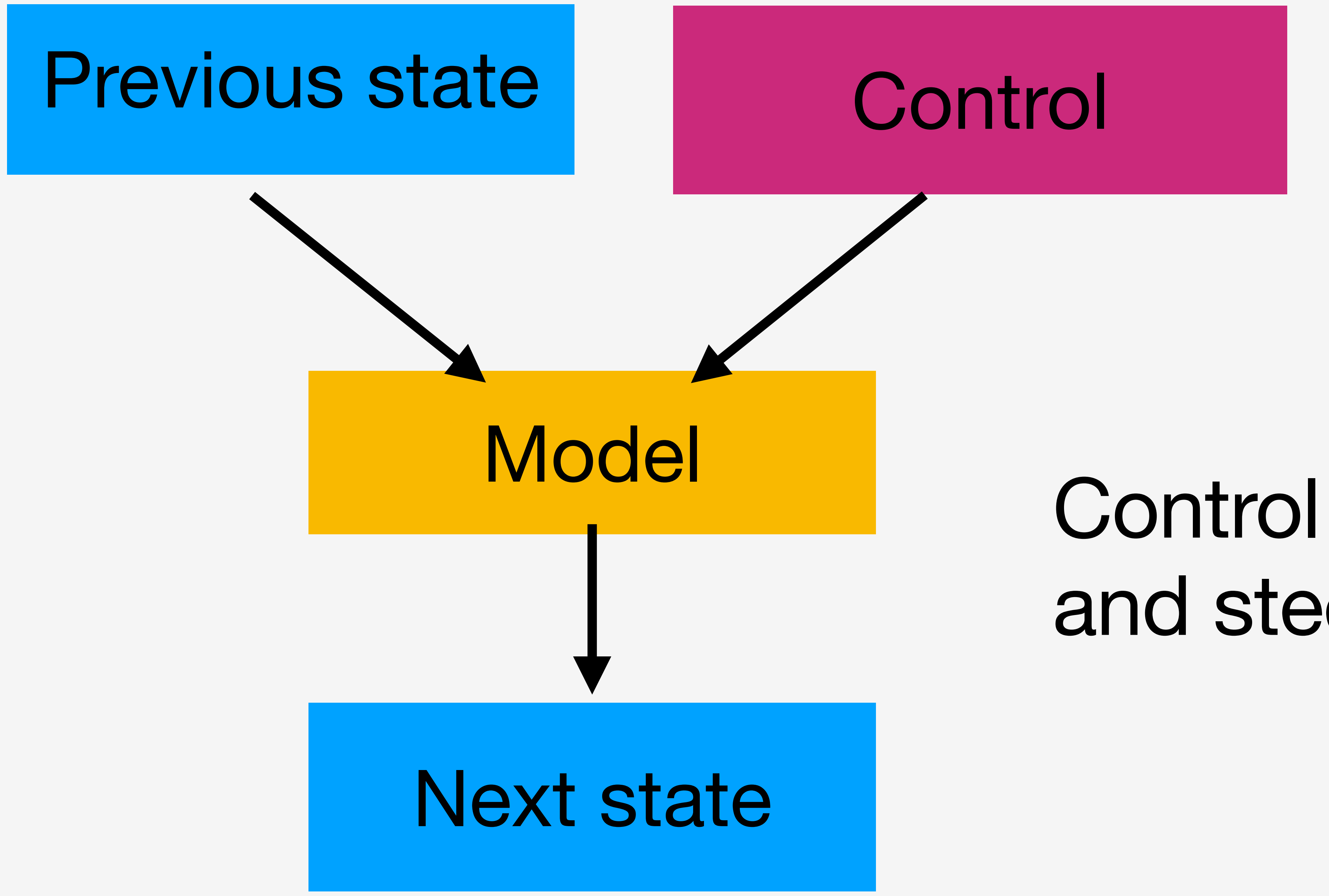
More simulations => less surprises

Kinematics may be adequate for slow-moving robots and relatively-flat terrains

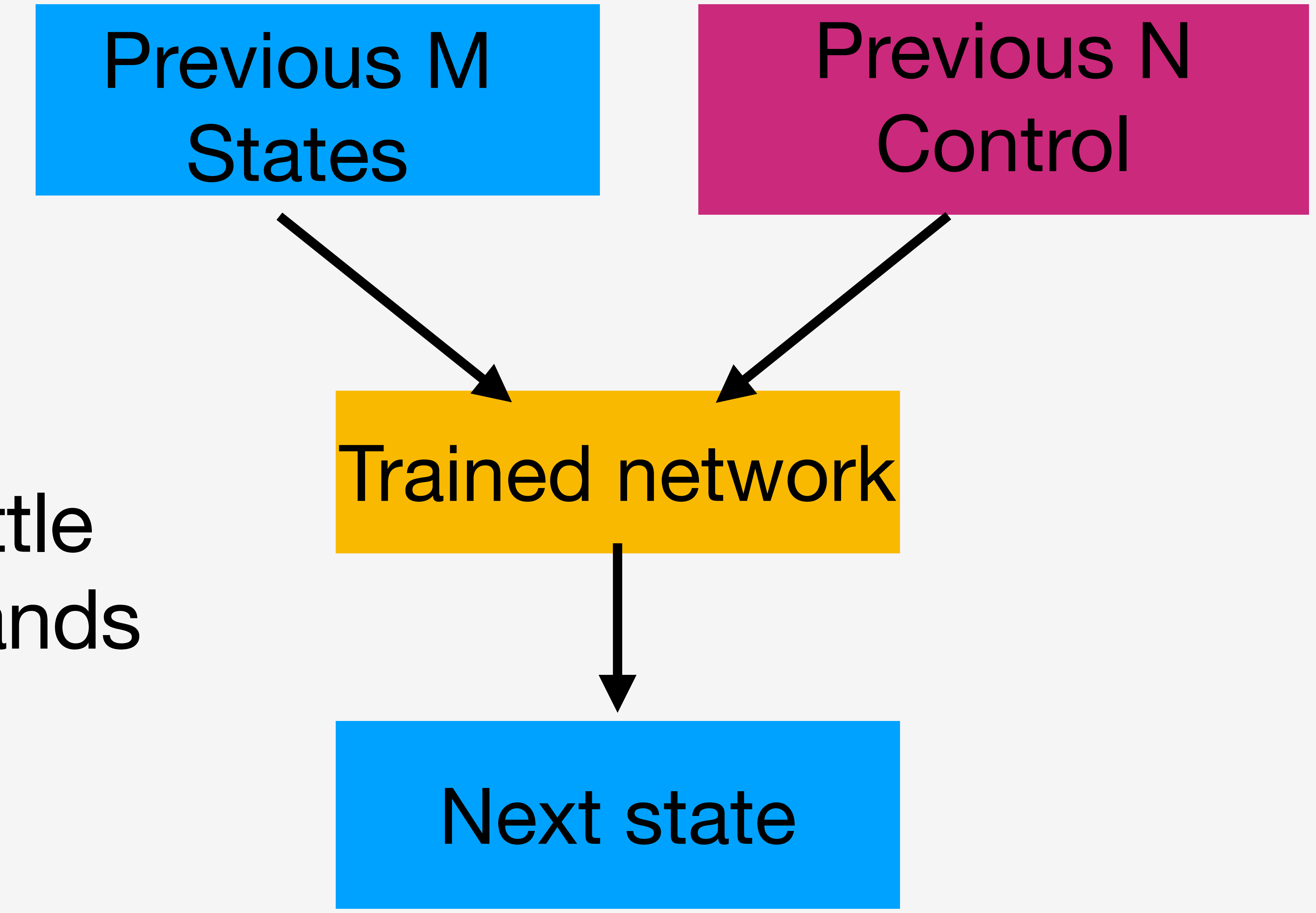
Simulating vehicular motion

ART PARK
AI & Robotics Technology Park

Model-driven



Data-driven



Control inputs : throttle and steering commands

Simple/ easy to interpret

Can learn arbitrarily complex machines

Unicycle Model

Point model for the Robot

Initially Robot is at A

Robot pose is (x, y, θ) in reference frame

Robot moves to B in time dt

Pose is now (x', y', θ')

Translation in terms of v

$$x' - x = v \cos\theta \, dt$$

$$y' - y = v \sin\theta \, dt$$

Rotation in terms of ω

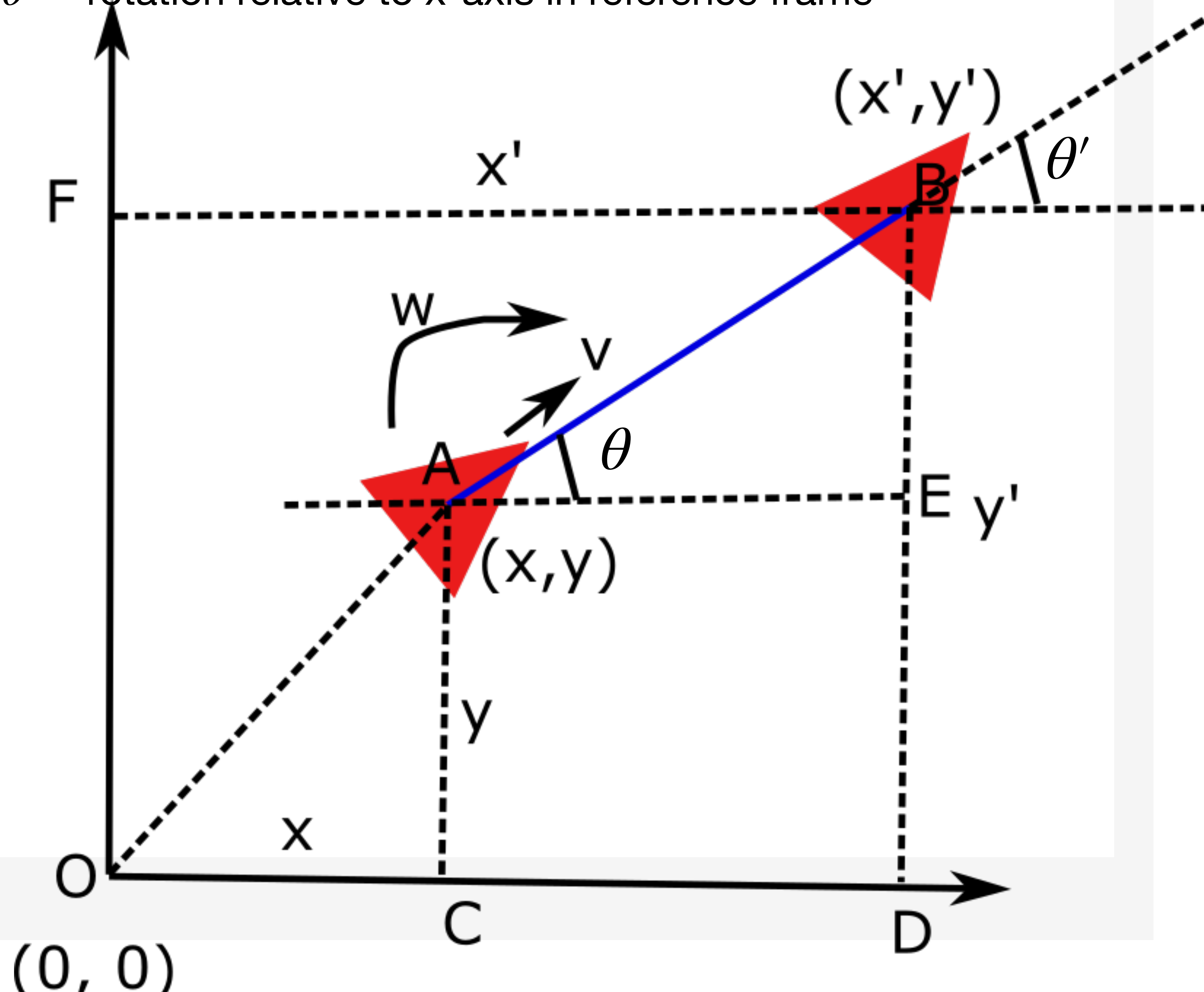
$$\theta' - \theta = \omega \, dt$$

- **v and ω are commands (cause)**
- **Change in robot pose is effect**

Robot pose

x, y = displacement with respect to origin in reference frame

θ = rotation relative to x-axis in reference frame



How to effect translation/ rotation?

Relation between v and ω

Point of contact with ground has no net force

- Wheel will get "dragged" otherwise

Wheel center moves same distance as any point in the rim (for ex: point A)

$$s = r\theta$$

Differentiating wrt time

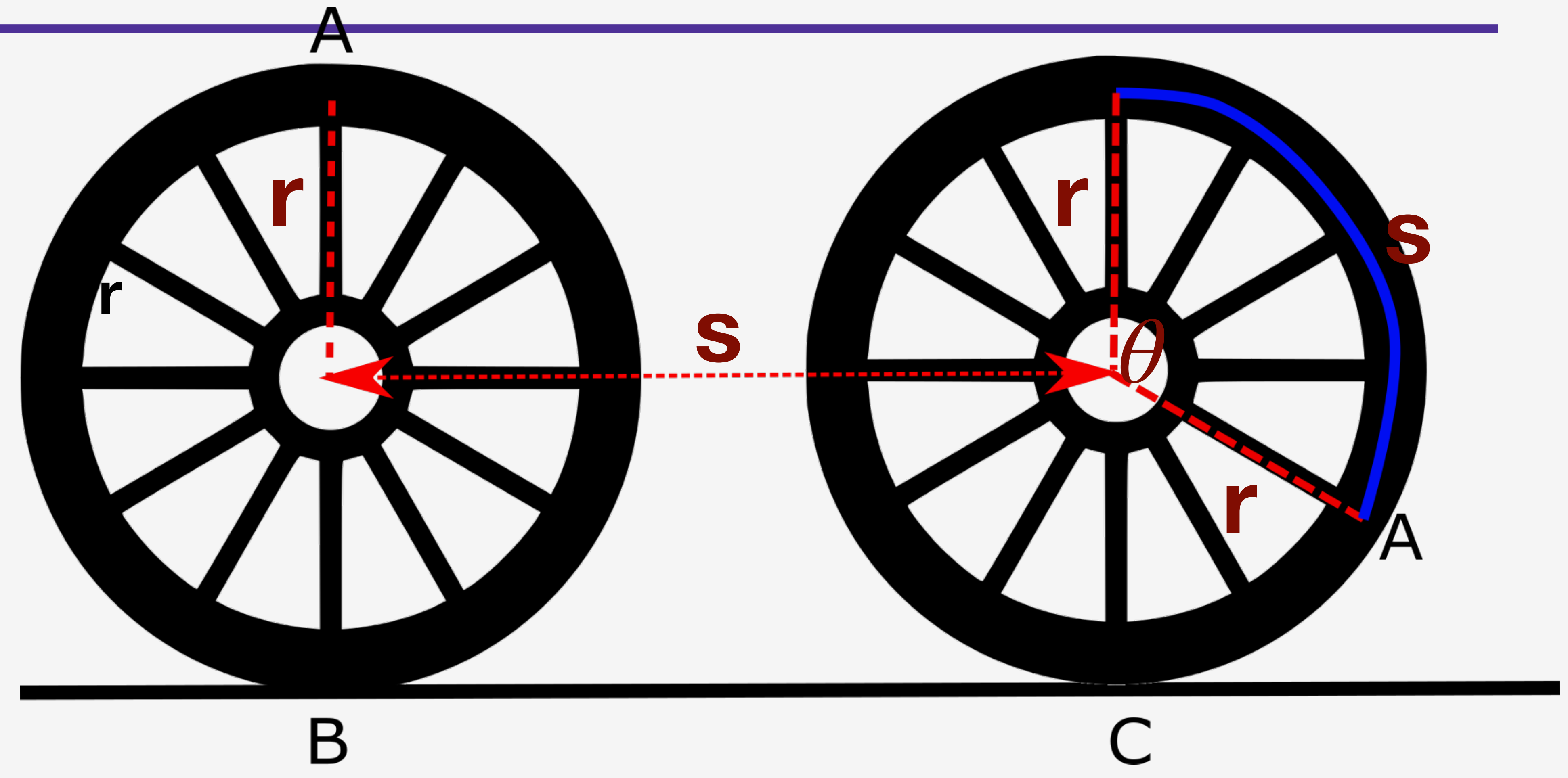
$$v = r\omega$$

Differential-drive robot

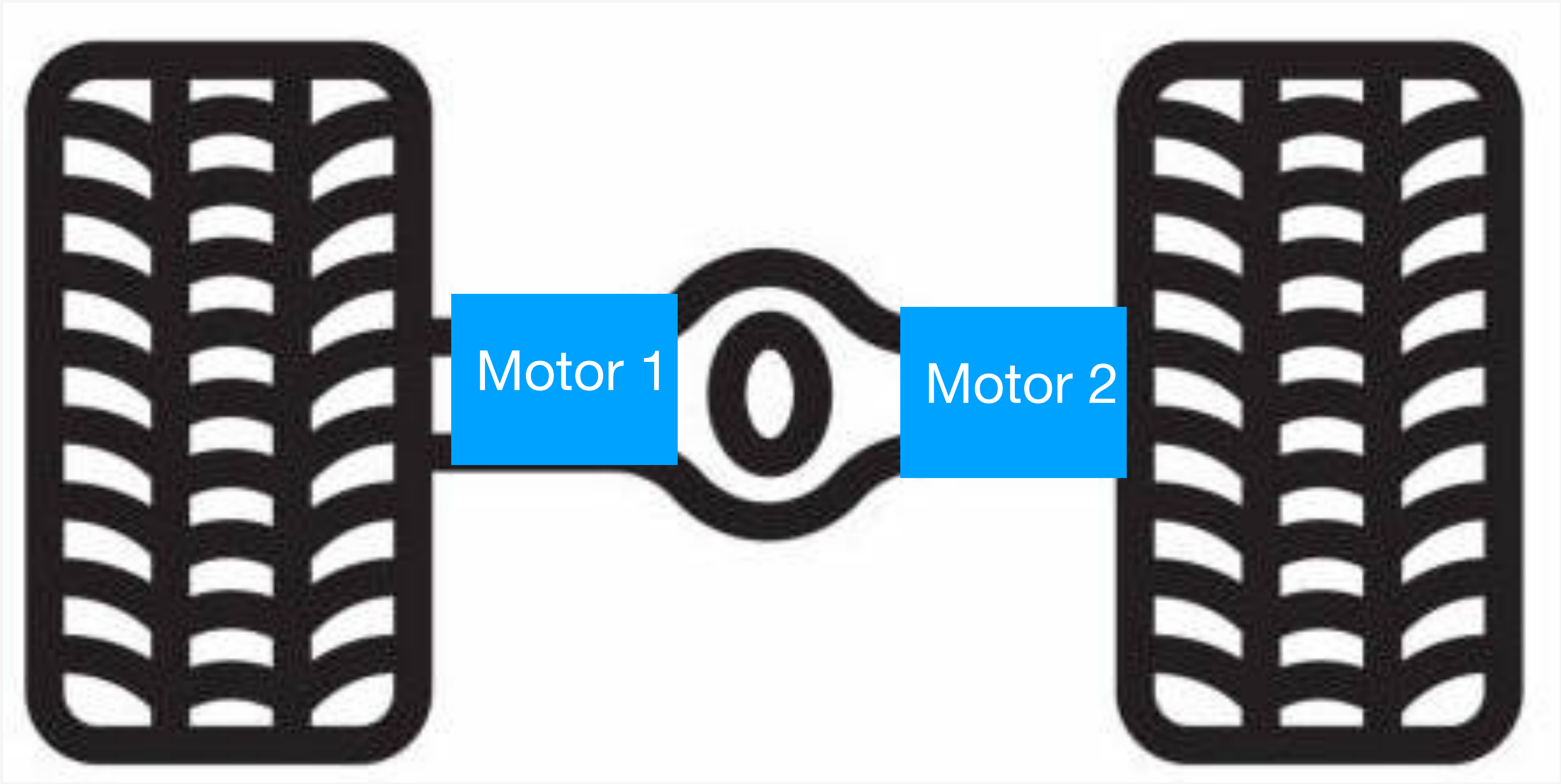
Connect 2 wheels through an axle

- Each wheel can be independently controlled with a motor

Roll without slipping



Question: where do you think Point B moved?



Differential Drive Model

Want the robot to take left turn

- Claim: Command left motor to rotate slower than right motor

Two rotations happening

- 2 wheels are rotating about the axle
- Rigid body (2 wheels + axle) rotates around an imaginary point - ICC

How are the 2 rotations related?

$$v_r = \omega (R + L/2)$$

$$v_l = \omega (R - L/2)$$

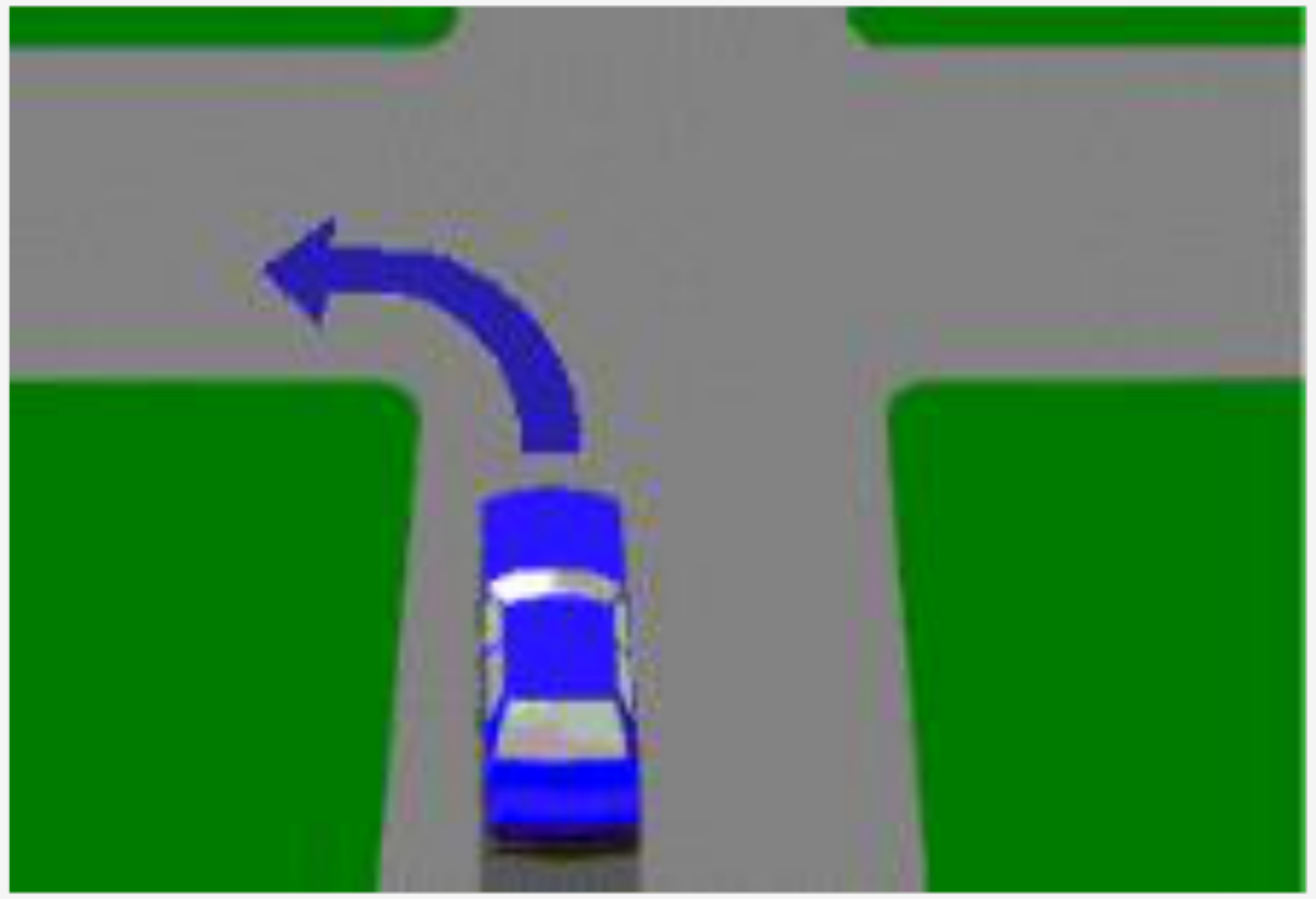
Adding,

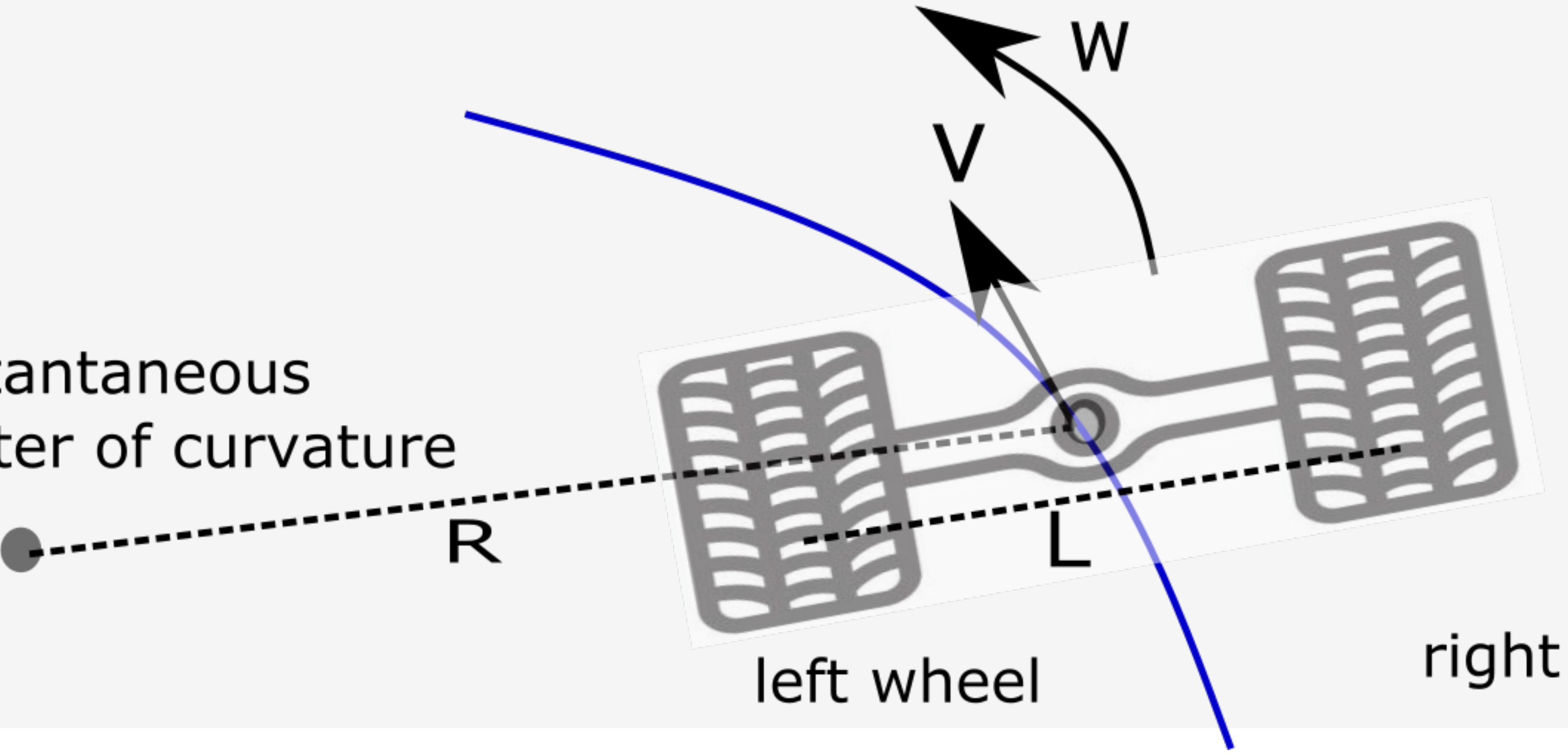
$(v_r + v_l)/2 = \omega R$ ~ avg velocity

Subtracting,

$(v_r - v_l)/L = \omega$

Vehicle takes left turn





Differential drive

Each wheel has own angular velocities

$$\omega_r = v_r r \qquad \omega_l = v_l r$$

Rewriting, v and ω for the system is

$$v = \frac{r}{2}(w_r + w_l)$$

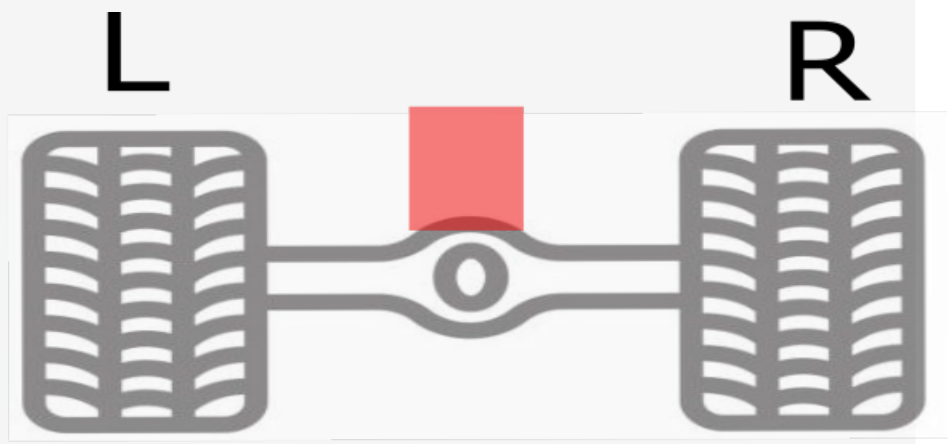
$$\omega = \frac{r}{L}(w_r - w_l)$$

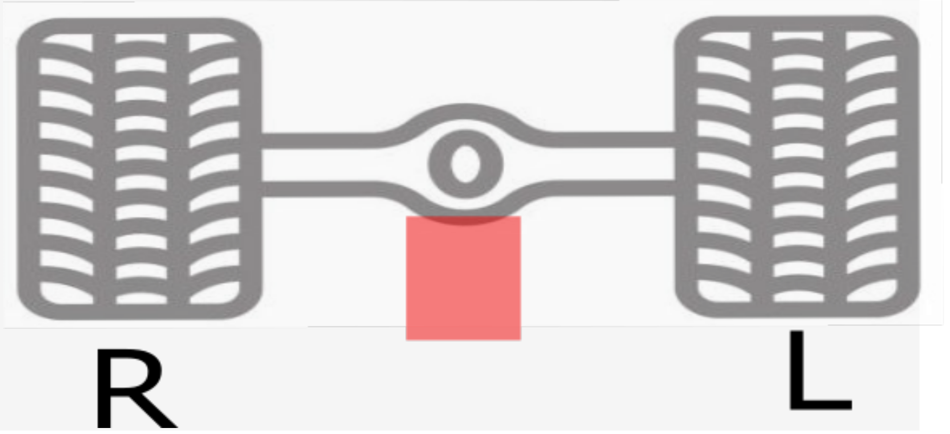
Curvature

$$R = \frac{L}{2}\frac{w_r + w_l}{w_r - w_l}$$

Interesting observations:

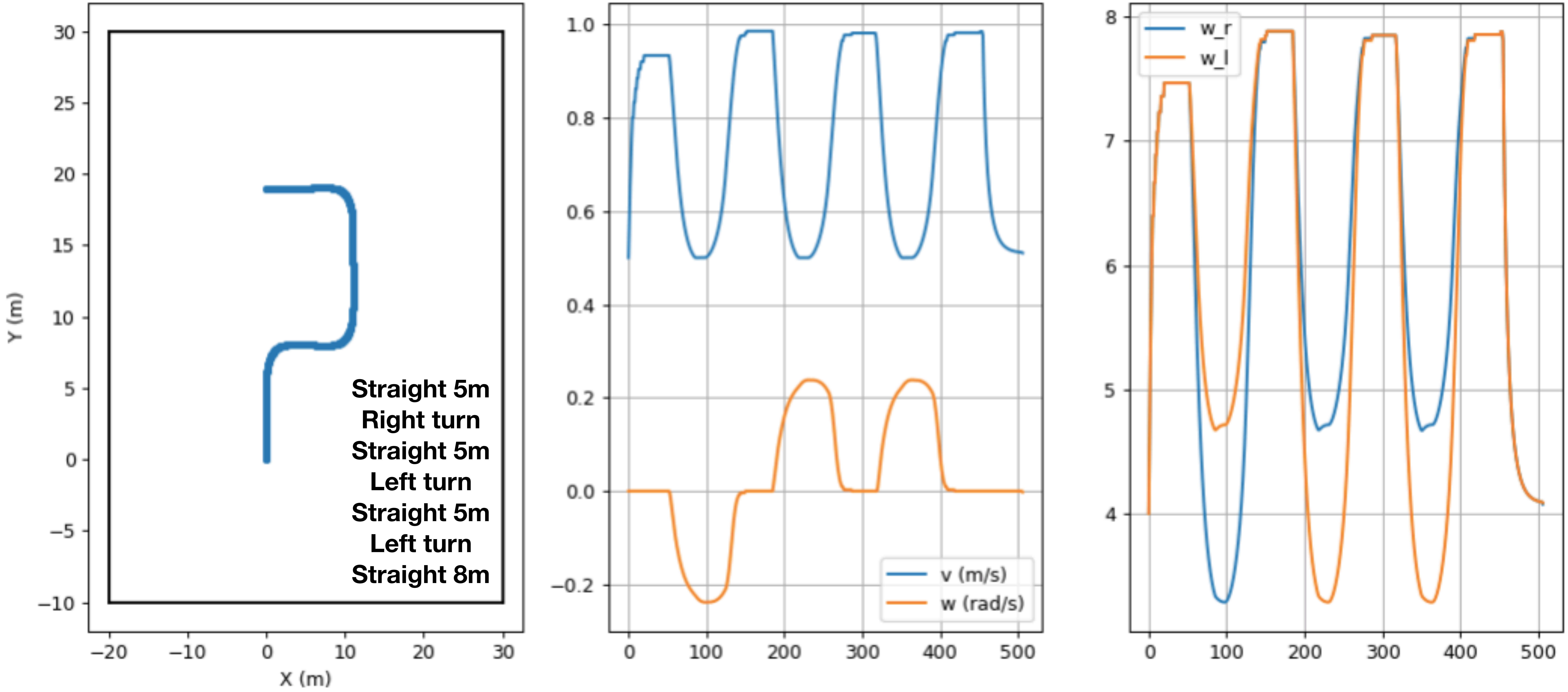
1. $\omega_r = \omega_l \implies$ Straight line
2. $\omega_r = -\omega_l \implies$ rotates about mid-point (Inplace)





Simulating Unicycle model

ART PARK
AI & Robotics Technology Park



1 m/s = 3.6 kmph

1 rad/s = 9.55 rpm

Remarks about simulation

Vehicle dimensions
- **Wheel radius = 0.125m**
- **Track width = 0.8m**

Turn: yaw change of 90 degrees
- **Cubic polynomial in time**
- **Omega is quadratic**

Reference path generated
- Function of path planner

v, ω **computed based on current robot pose**
- Function of path tracker

Current pose of robot computed using vehicle model
- Unicycle/ Bicycle model

Velocity explicitly shaped
- **Start from rest**
- **Slow down exponentially when parking**

Closed-loop simulation of actual vehicle

Further Discussion

1. Can you make a rough estimate how long it takes to complete the 4 straight segments?

(a) 20 seconds (b) 15 seconds (c) 30 seconds (d) 25 seconds

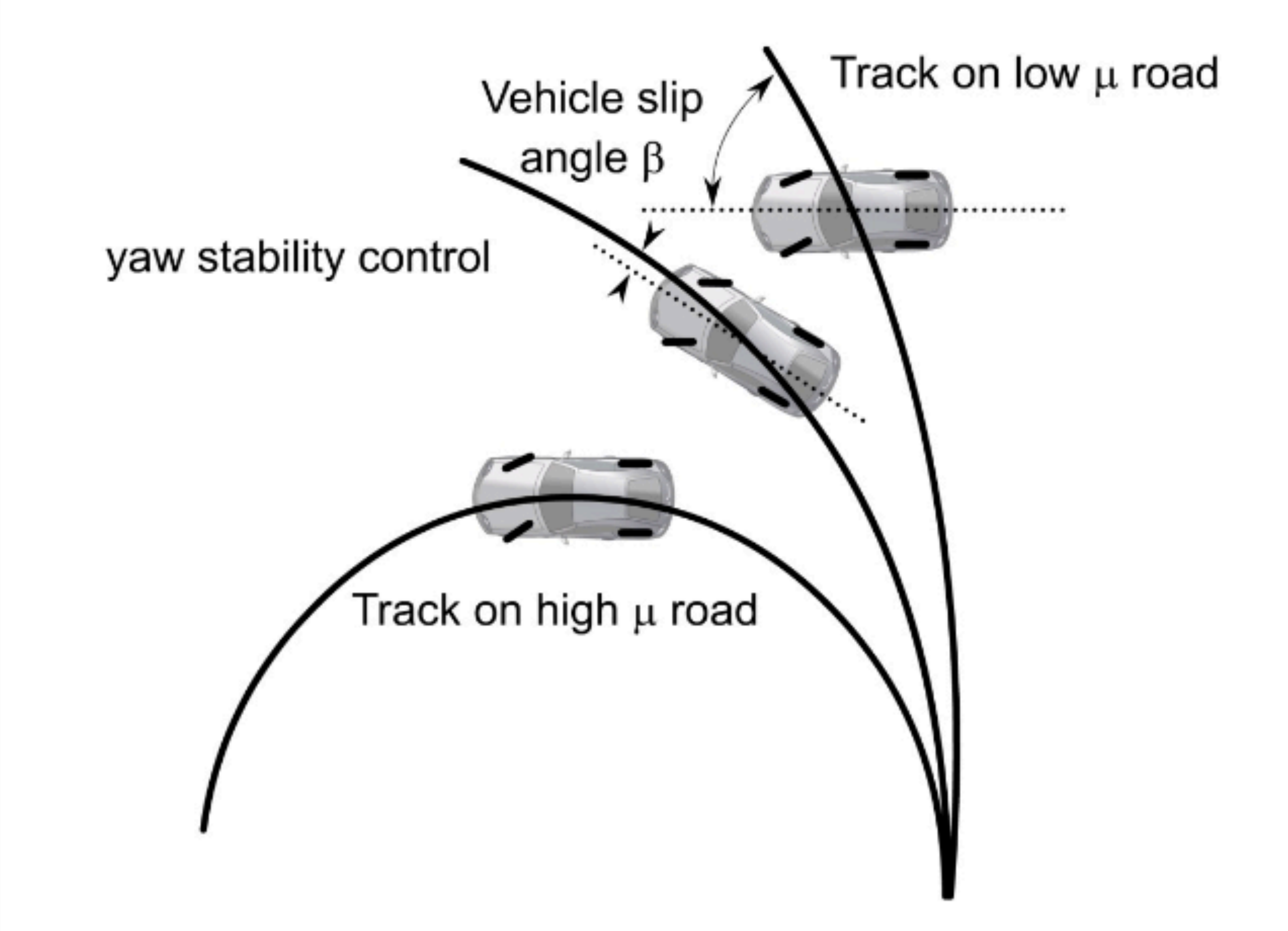
2. Can you make a rough estimate how long it takes to complete the 3 turn segments?

(a) 20 seconds (b) 15 seconds (c) 10 seconds (d) 25 seconds

3. For robots running at 15 Kmph, what kind of wheel speeds can we expect?

(a) 280-300 rpm (b) 350-370 rpm (c) 200-220 rpm (d) 150-170 rpm

Need for Yaw control system



R. Rajamani, Vehicle Dynamics and Control, Chapter 2, Mechanical Engineering Series

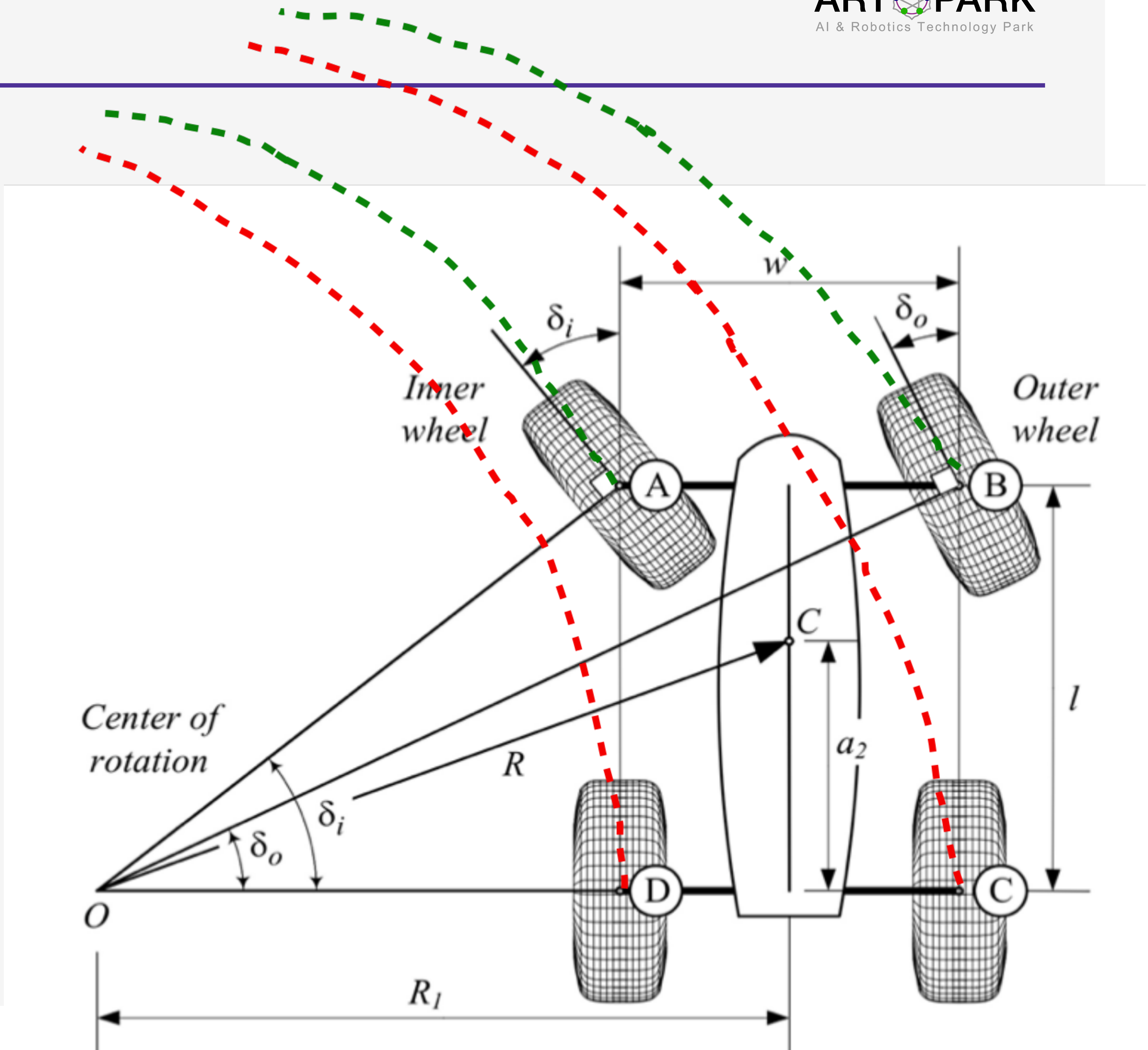
Ackerman steering

Prevent skidding at high speeds

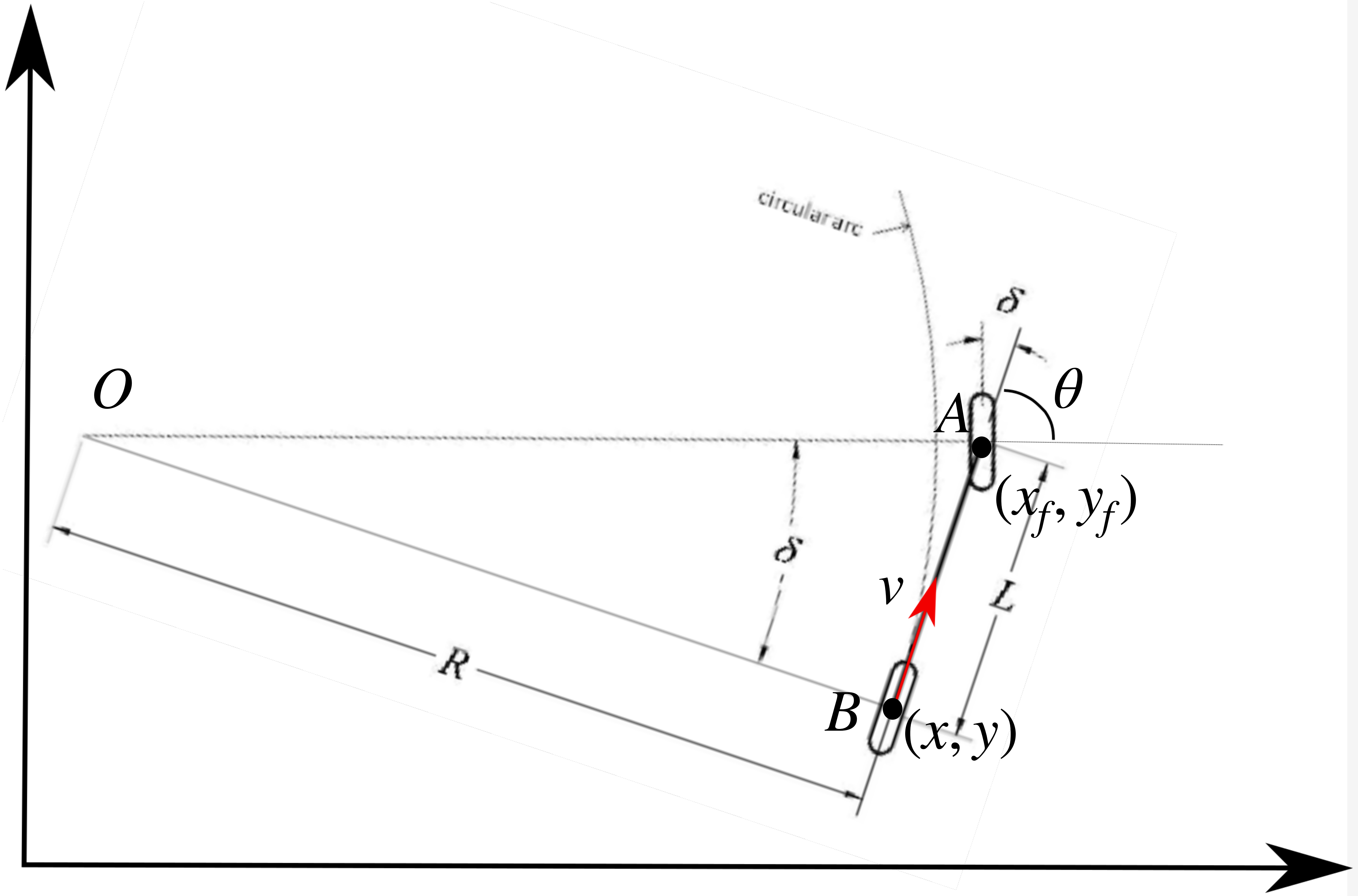
All 4 wheels rotate around a common center

- Right wheels rotate faster than left wheels (for a left turn)
- Front wheel rotate faster than back wheel

$$\omega_B > \omega_C > \omega_A > \omega_D$$



Bicycle Model



Front wheels and Rear wheels separately lumped

Control inputs:

- Front wheel can be steered (δ)
- Rear wheel is powered (v)

$$\dot{x} = v \cos\theta$$

$$\dot{y} = v \sin\theta$$

$$\dot{\theta} = \frac{v}{L}\tan\delta$$

Bicycle model

ARTPARK
AI & Robotics Technology Park

Derivation

No sideways motion for both front and rear wheels

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0$$ - (1)

$$\dot{x} \sin\theta - \dot{y} \cos\theta = 0$$ - (2)

Because it is a rigid body

$$x_f = x + L\cos\theta$$ - (3)

$$y_f = y + L\sin\theta$$ - (4)

Using (3) and (4) in (1)

$$\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - \dot{\theta} L \cos\delta = 0$$

(2) can be satisfied by

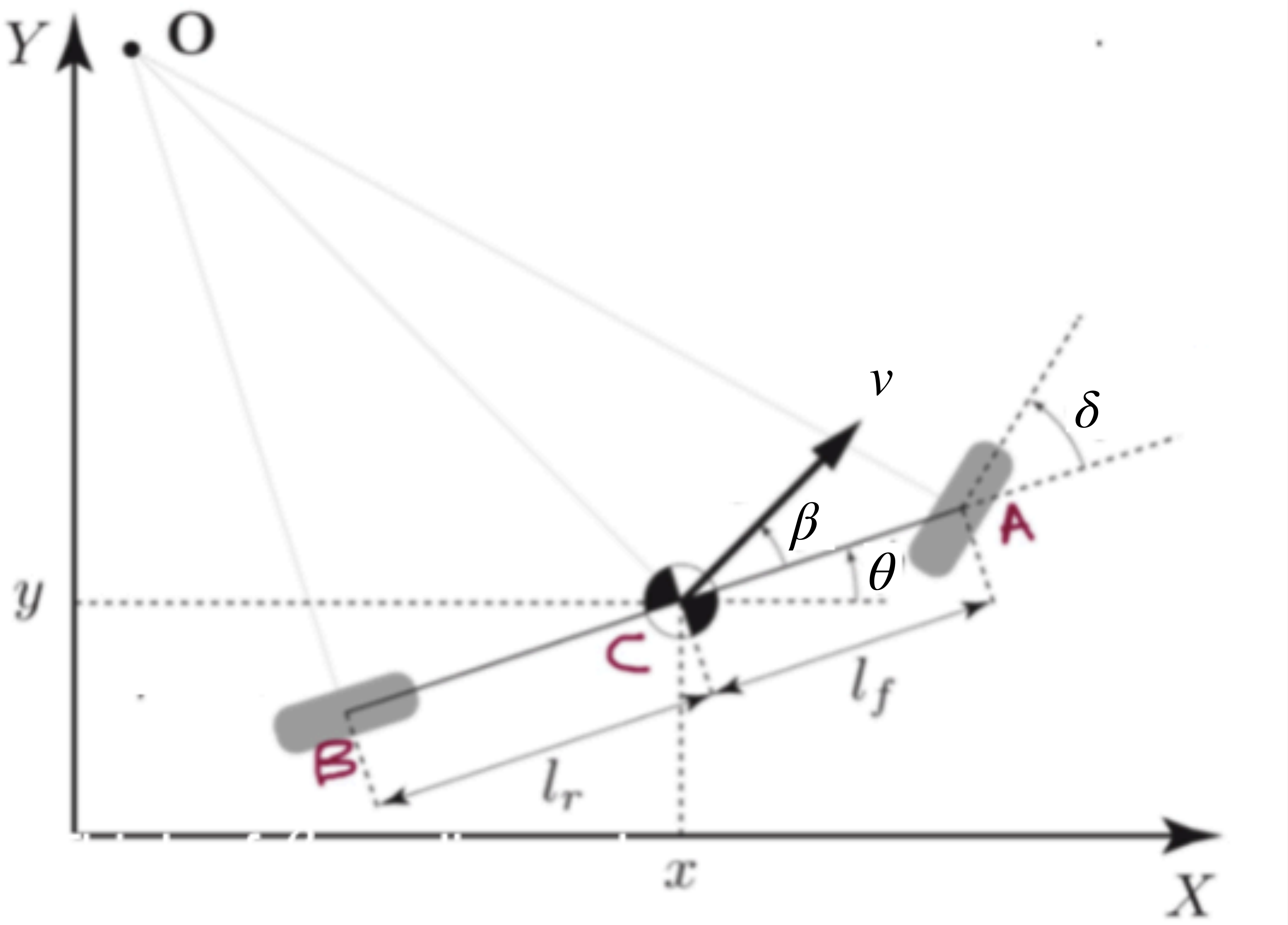
$$\dot{x} = v\cos\theta \quad \dot{y} = v\sin\theta$$

We can then show

$$\dot{\theta} = \frac{v\tan\delta}{L}$$

$$R = \frac{v}{\dot{\theta}} = \frac{L}{\tan\delta}$$

Bicycle Model
Kinematics with side-slip



Think of β as slip angle

Vehicle is being pushed sideways

A, B = location of front and rear wheels

C = location of vehicle center-of-gravity

$$\dot{x} = v\cos(\theta + \beta)$$

$$\dot{y} = v\sin(\theta + \beta)$$

$$\dot{\theta} = \frac{v}{l_f + l_r}\cos\beta\tan\delta$$

$$\tan\beta = \frac{l_r}{l_r + l_f}\tan\delta$$

When does β become important?
(a) High friction
(b) High Speed
(c) Low Friction
(d) Low Speed

Bicycle model with side-slip

Derivation

Let radius of rotation, i.e OC = R

In Δ OAC, from law of sines

$$\sin(\delta - \beta)/l_f = \cos(\delta)/R$$

$$\implies \tan\delta\cos\beta - \sin\beta = l_f/R \quad - \ (1)$$

In Δ OBC, using law of sines

$$\sin\beta = l_r/R \quad - (2)$$

(1) + (2) gives $\tan\delta\,\cos\beta = (l_f + l_r)/R$ - (3)

Using $v = R\dot{\theta}$ in (3)

$$\dot{\theta} = \frac{v}{l_f + l_r}\cos\beta\tan\delta$$

(2) / (3) gives

$$\tan\beta = \frac{l_r}{l_r + l_f}\tan\delta$$

Recommended resources

1. Vehicle Dynamics and Control by Prof. Georg Schildbach

3. https://m.youtube.com/watch?v=eblPwXFb7TE interview with Eberhard and Tarpenning

4. Martin Engelhardt, Control of Mobile Robots course lectures, Georgia Tech

5. Prof. Krishnakumar, Vehicle Dynamics, IIT-Madras, NPTEL

ARTPARK
AI & Robotics Technology Park