Introduction to Autonomous Electric Vehicles

Lecture 3

- Recap Lecture 2
- Introduction to sensors
- Wheel encoders
- Lidar

EEE department, BITS-Goa

31-March-2021

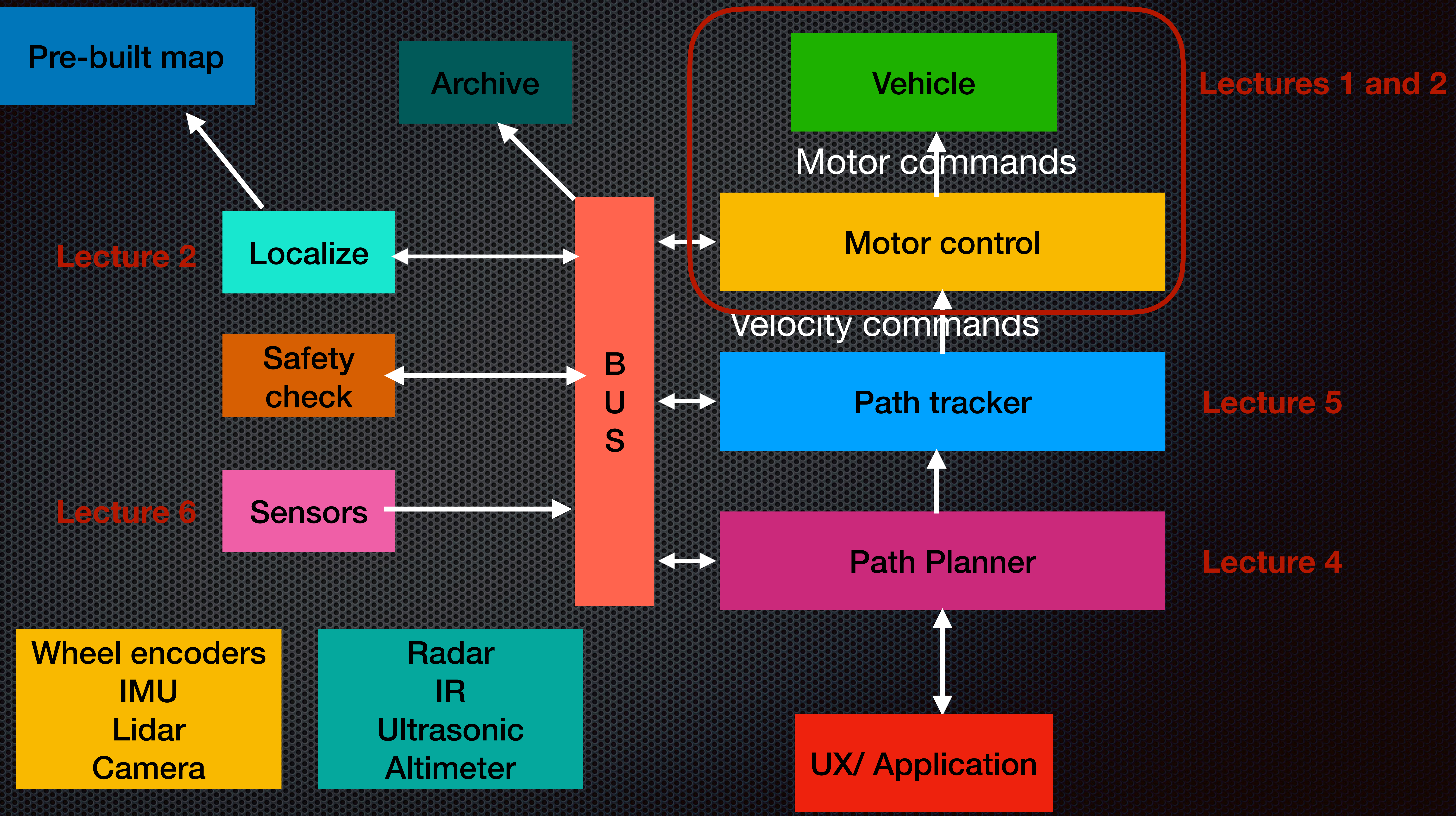
Lecture plan

- Recap Lec 2
  - Vehicle Modeling
  - Autonomy and control architecture
  - Need for Closed-loop feedback
- Introduction to Sensors
  - Incremental wheel encoders
    - Vehicle drift
  - Lidar
    - 2D lidar simulator
- Sensor Noise
- Simultaneous Mapping and Localization
  - What is localization?
  - Localization using Lidars
  - Sensor fusion and Kalman filters
  - Drift and Loop closure in Mapping

Autonomy and Control: Pub-Sub architecture

Iterative procedure

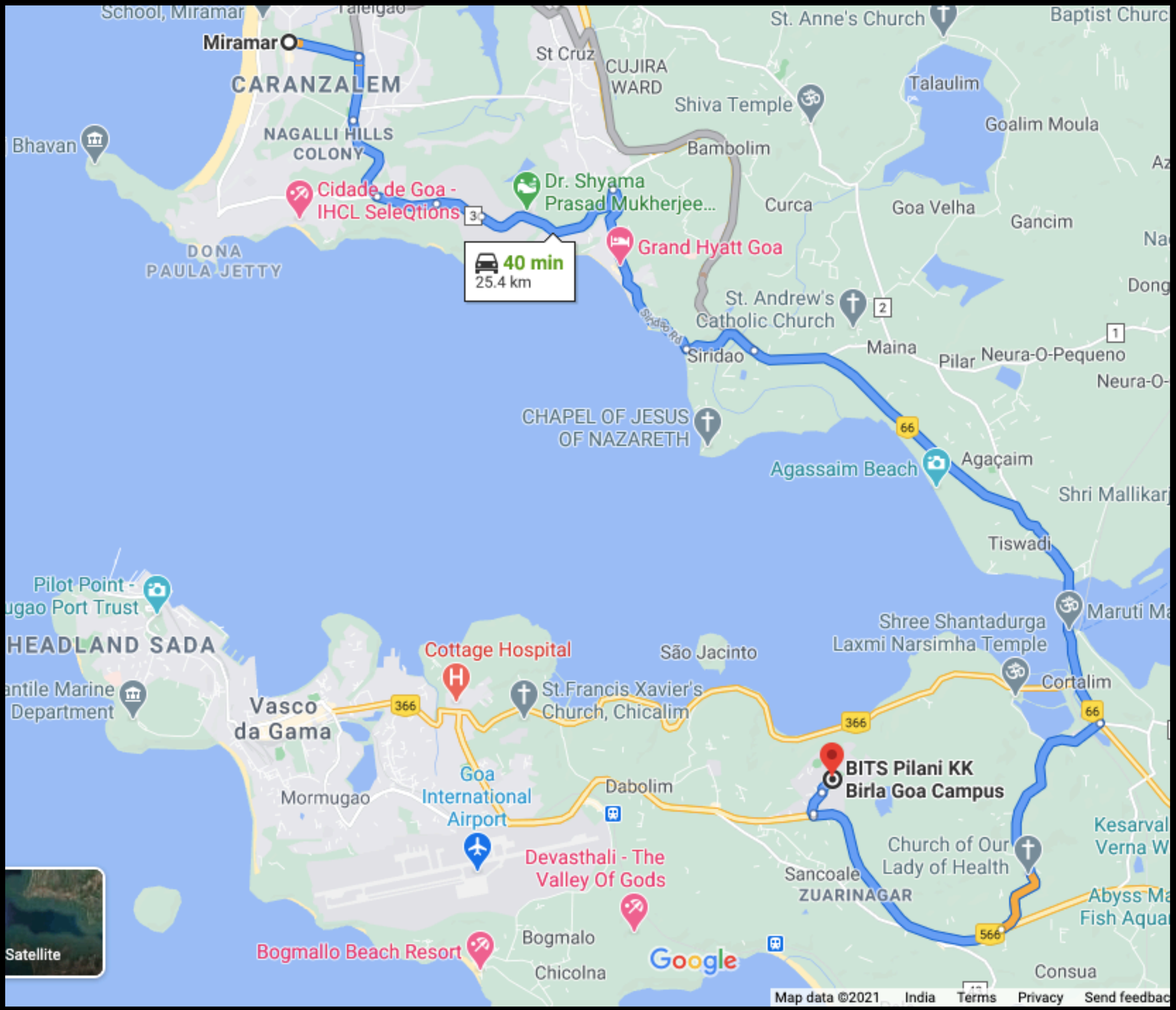
1. Where am I?
2. Where do I go next?
3. What should I do?



Key assumption: Operation in known environments

Maps and planning

Global planning: Go from campus to Miramar



Useful for High-level estimates

Locally safe planning

Vehicle action in immediate time horizon

- More annotations/ Semantically richer
- Suited for quick Replanning

Unicycle Model

Initially Robot is at A

In reference frame robot pose is (x, y, θ)

At velocity v, Robot moves to B in time dt

Pose is now (x', y', θ')

Relative motion in terms of v

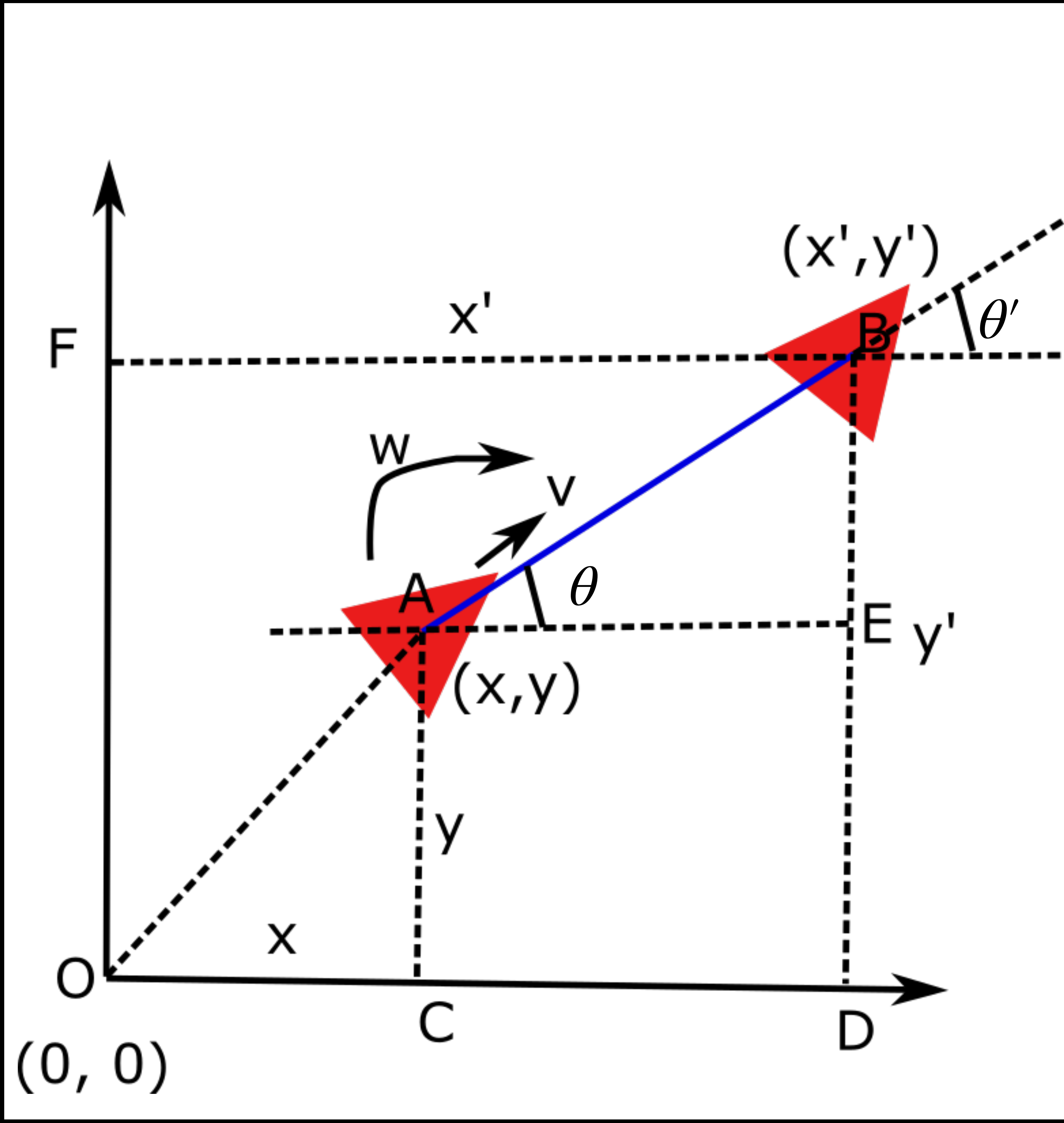
$$x' - x = v \cos\theta \, dt$$

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

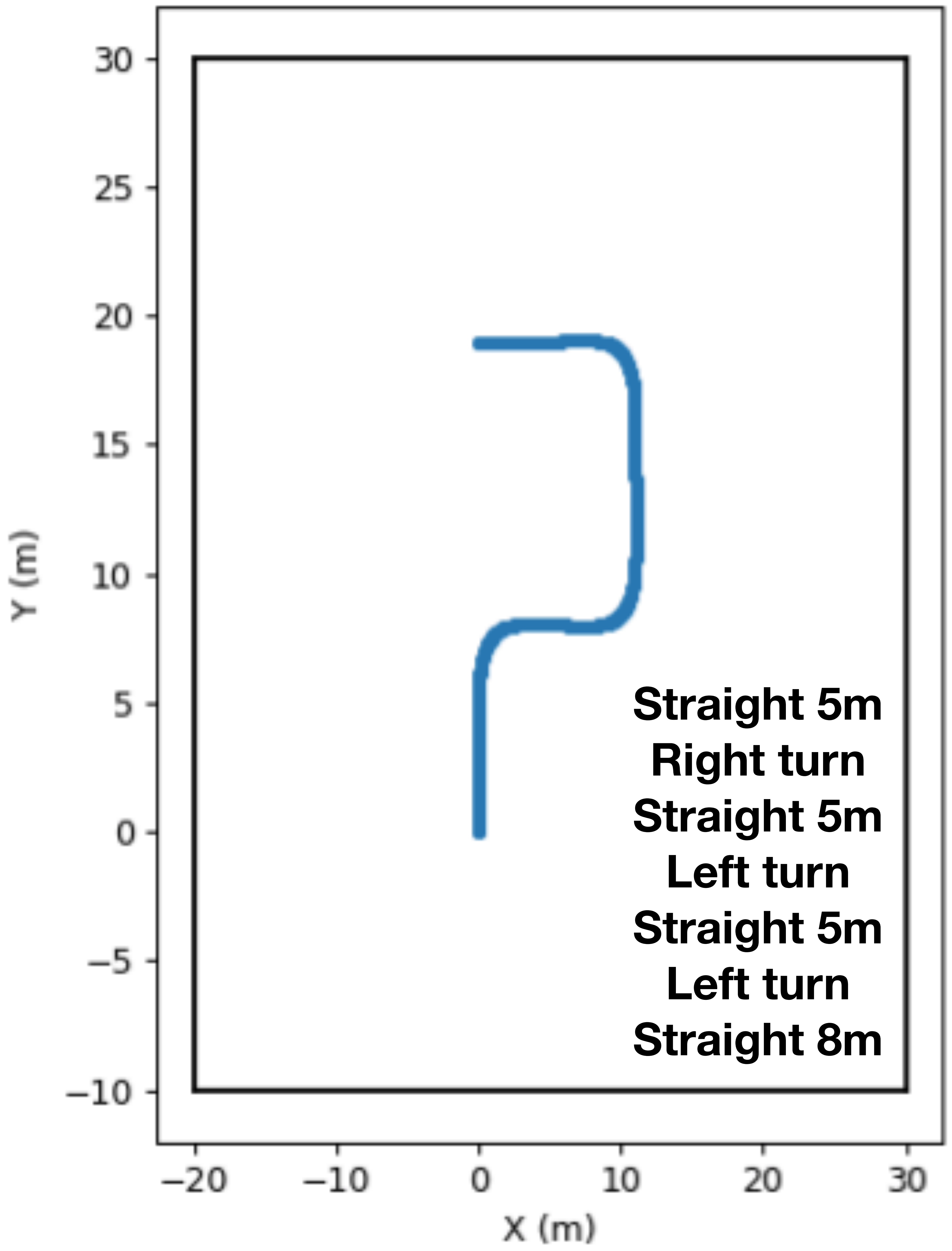
If robot also rotates at ω

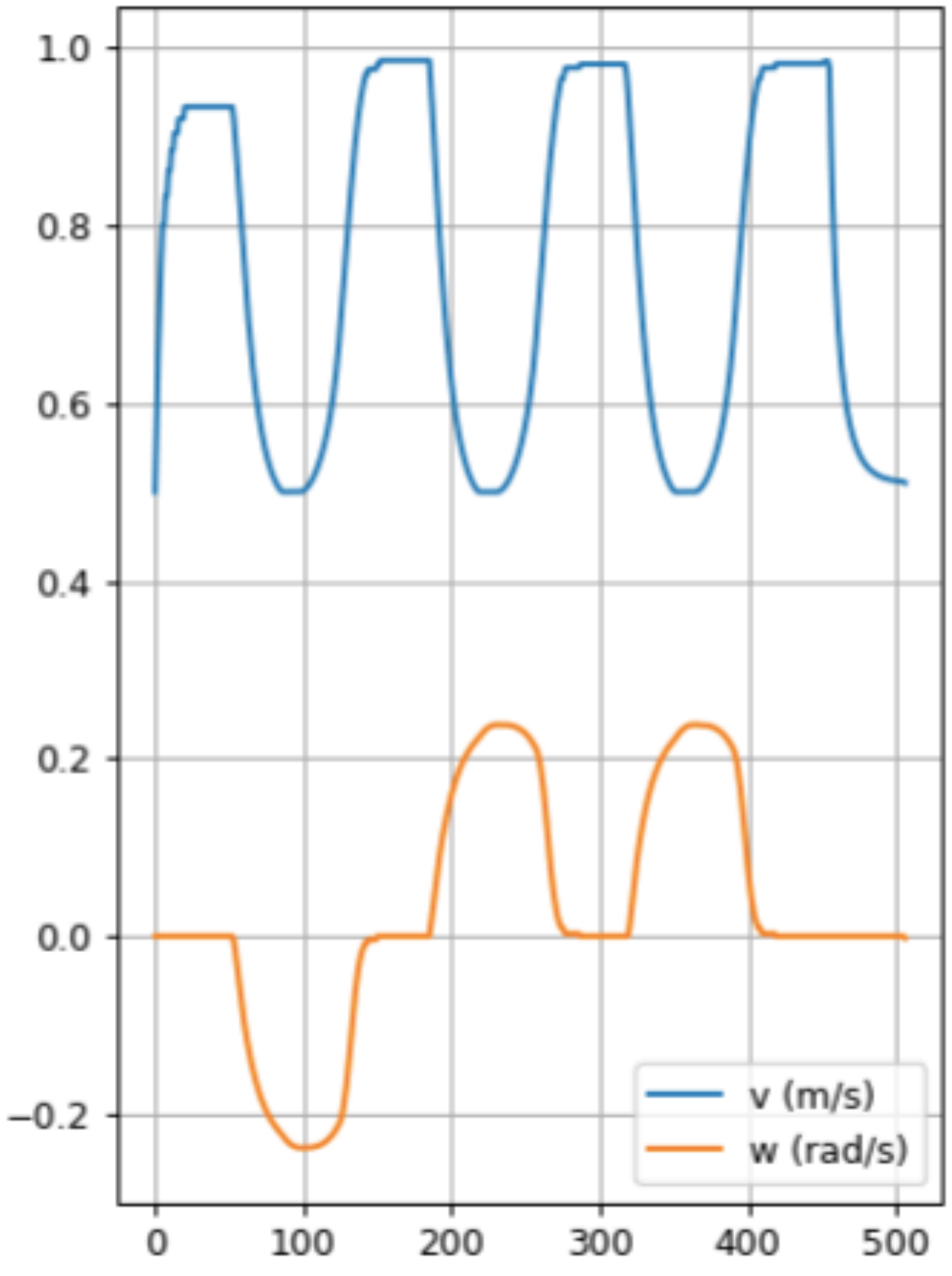
$$\theta' - \theta = \omega \, dt$$

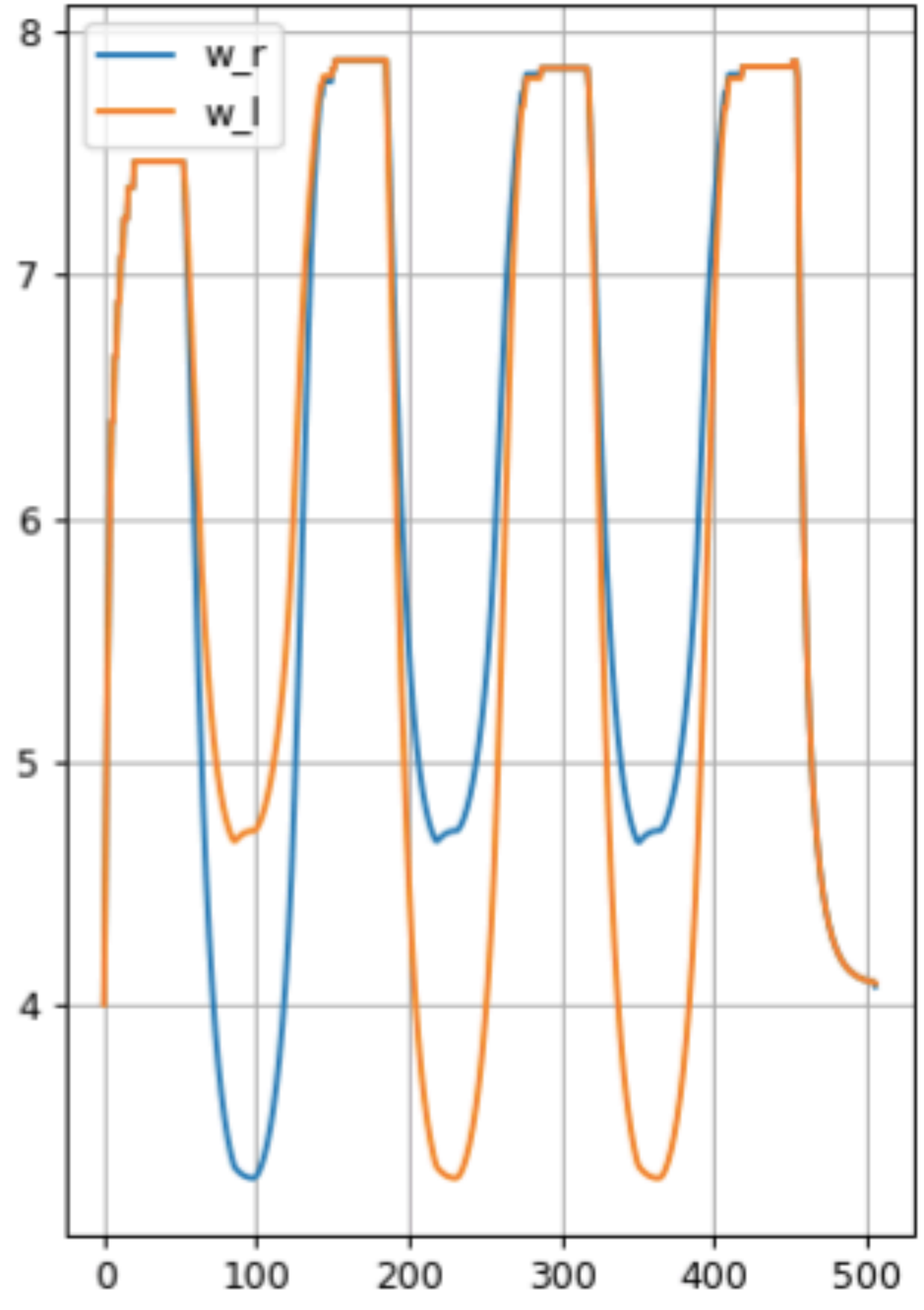
- **v and ω are commands**
- **Change in robot pose is effect**



Vehicle trajectory example







1 m/s = 3.6 kmph

1 rad/s = 9.55 rpm

Differential-drive model

Point of contact with ground has no net force

- Wheel will get "dragged" otherwise

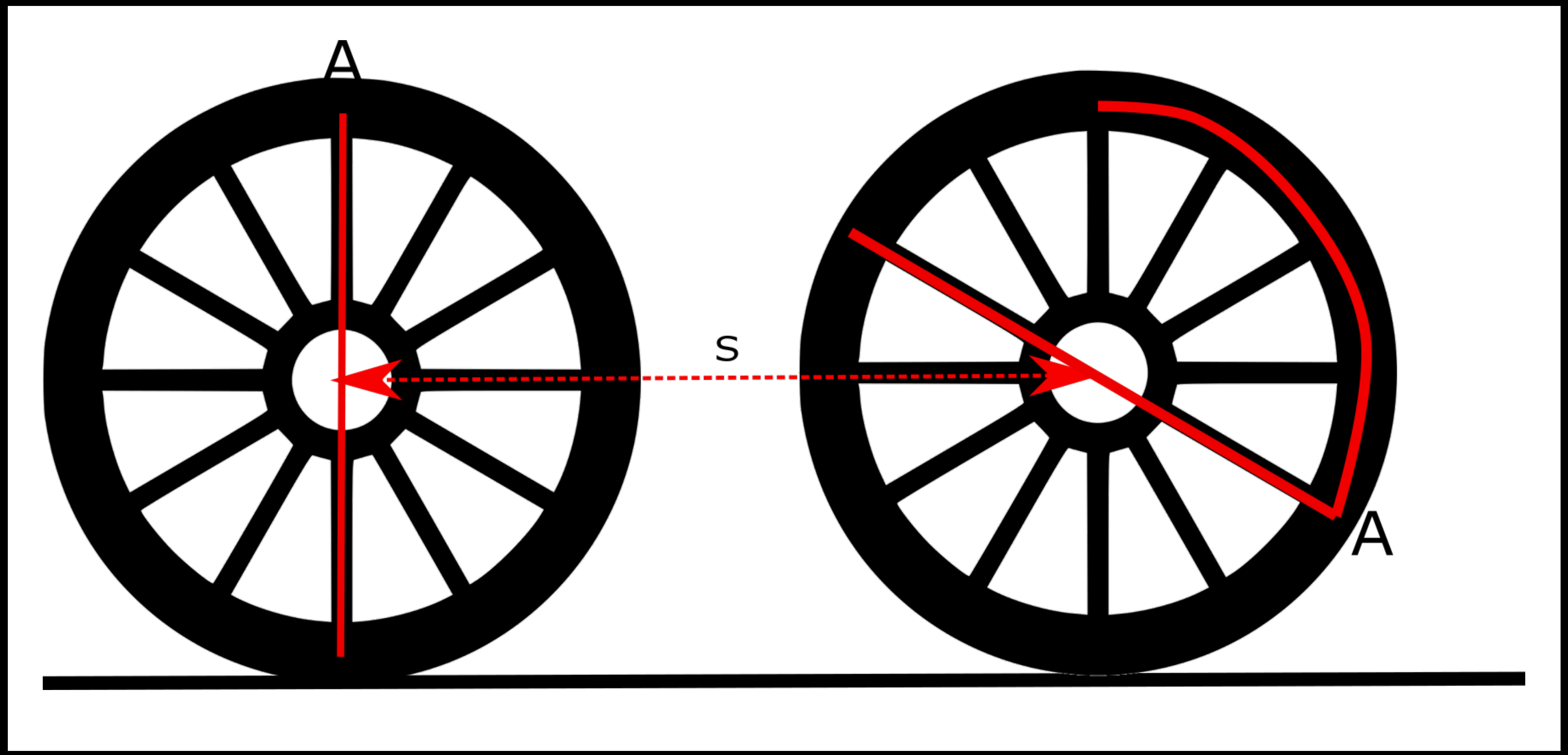
Wheel center moves same distance as any point in the rim (for ex: point A)

$$s = r\theta$$

Differentiating wrt time

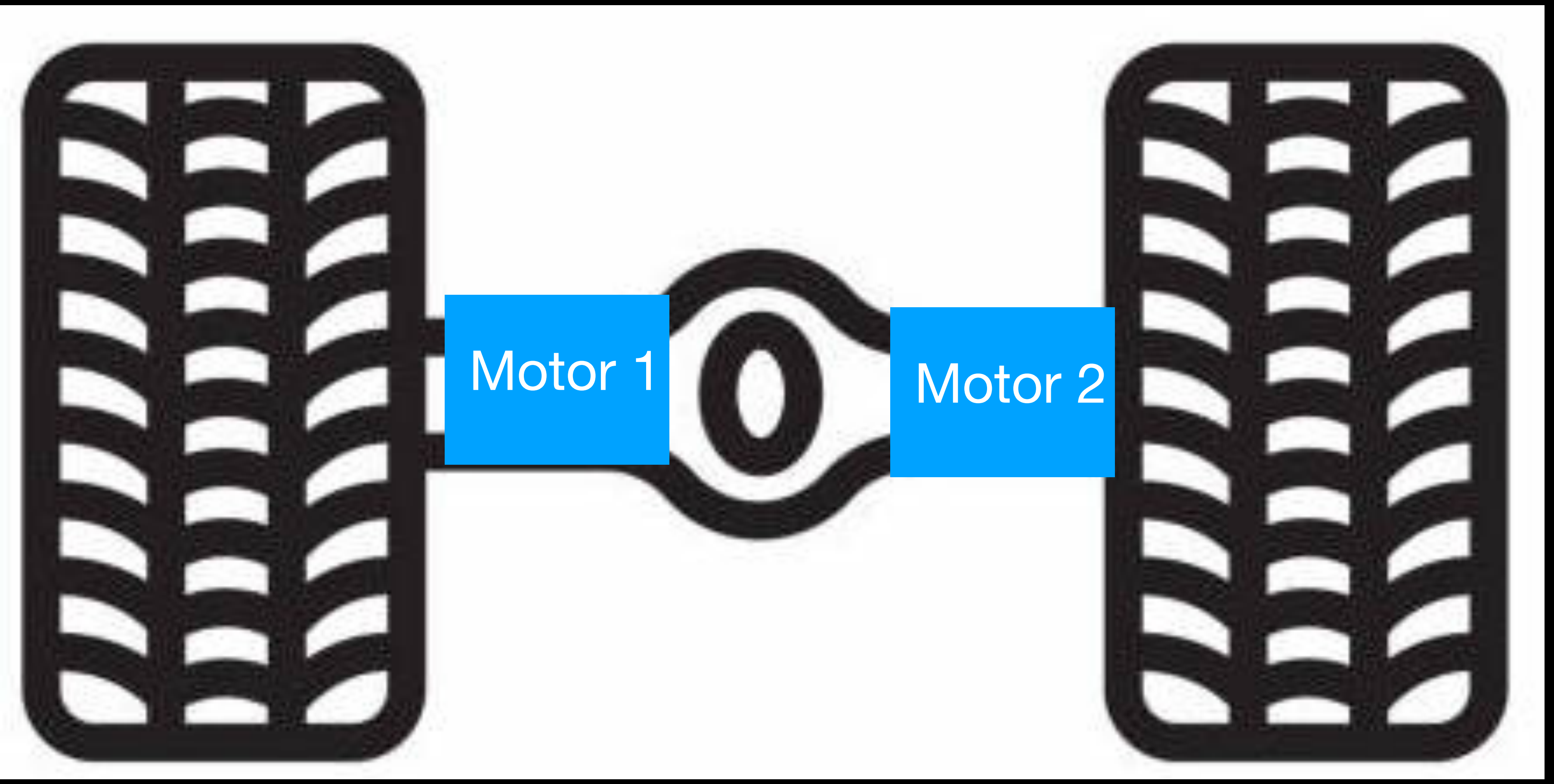
$$v = r\omega$$

Roll without slipping

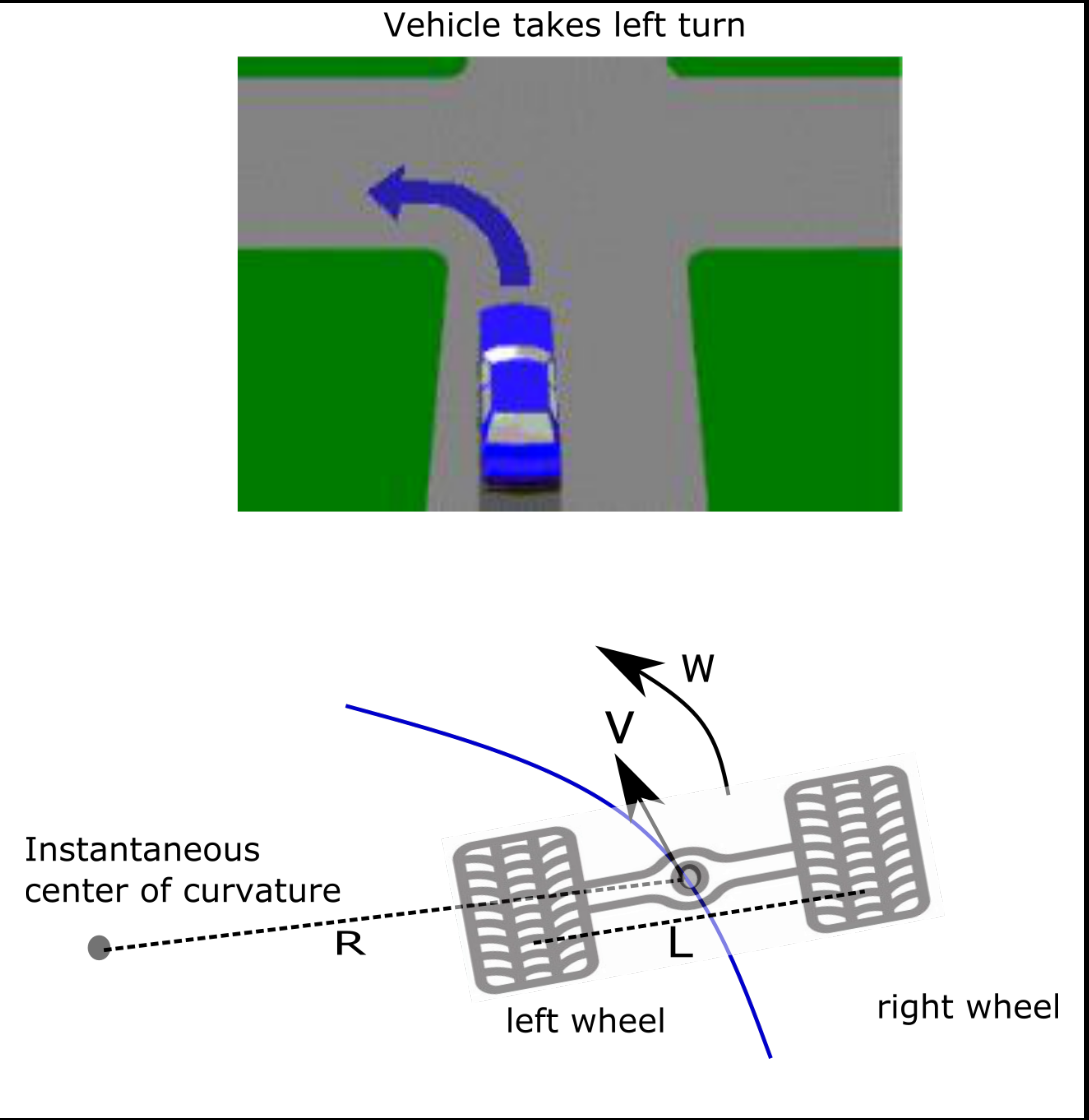


Simple robot

- Connect 2 wheels through an axle
- Each wheel can be independently controlled with a motor



Differential Drive Model



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

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

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

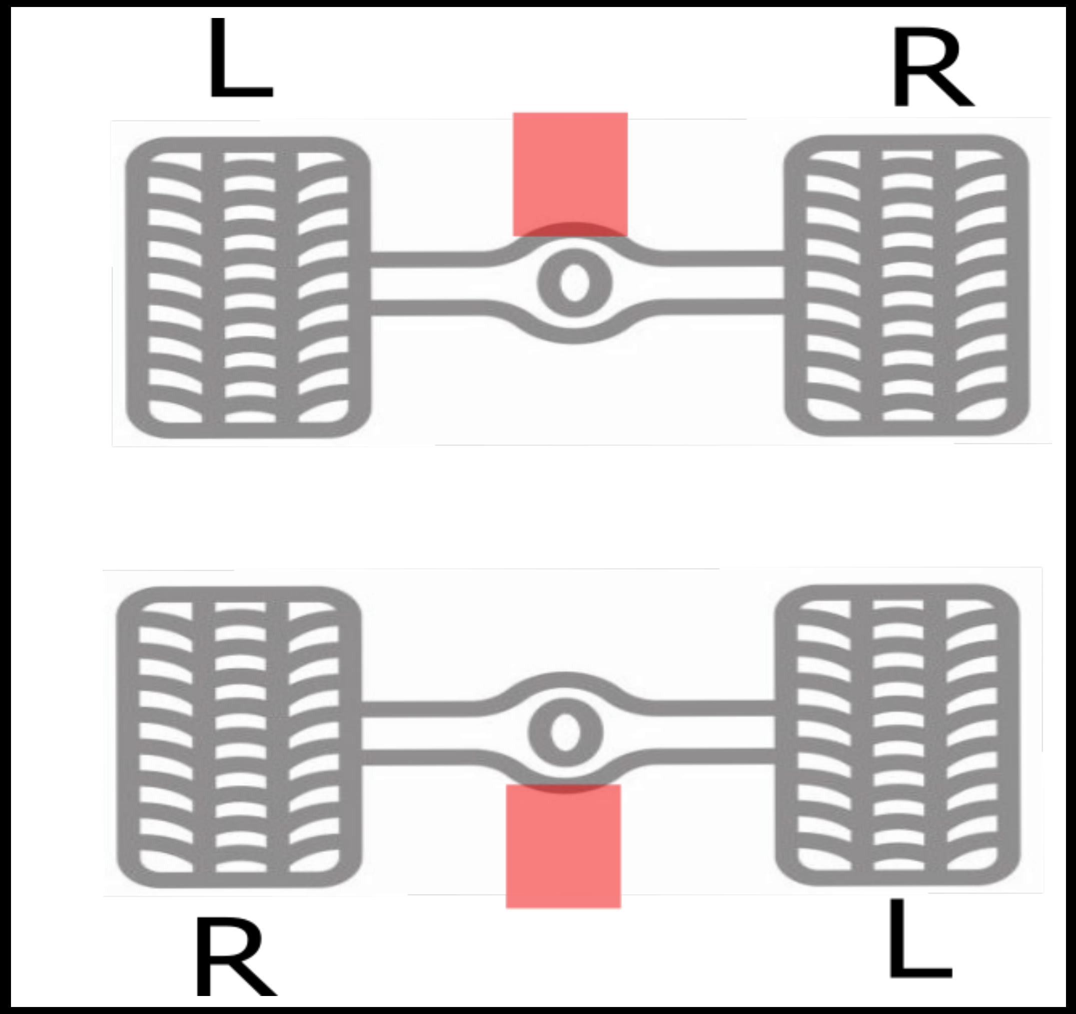
How to realize desired v, ω ?

Curvature

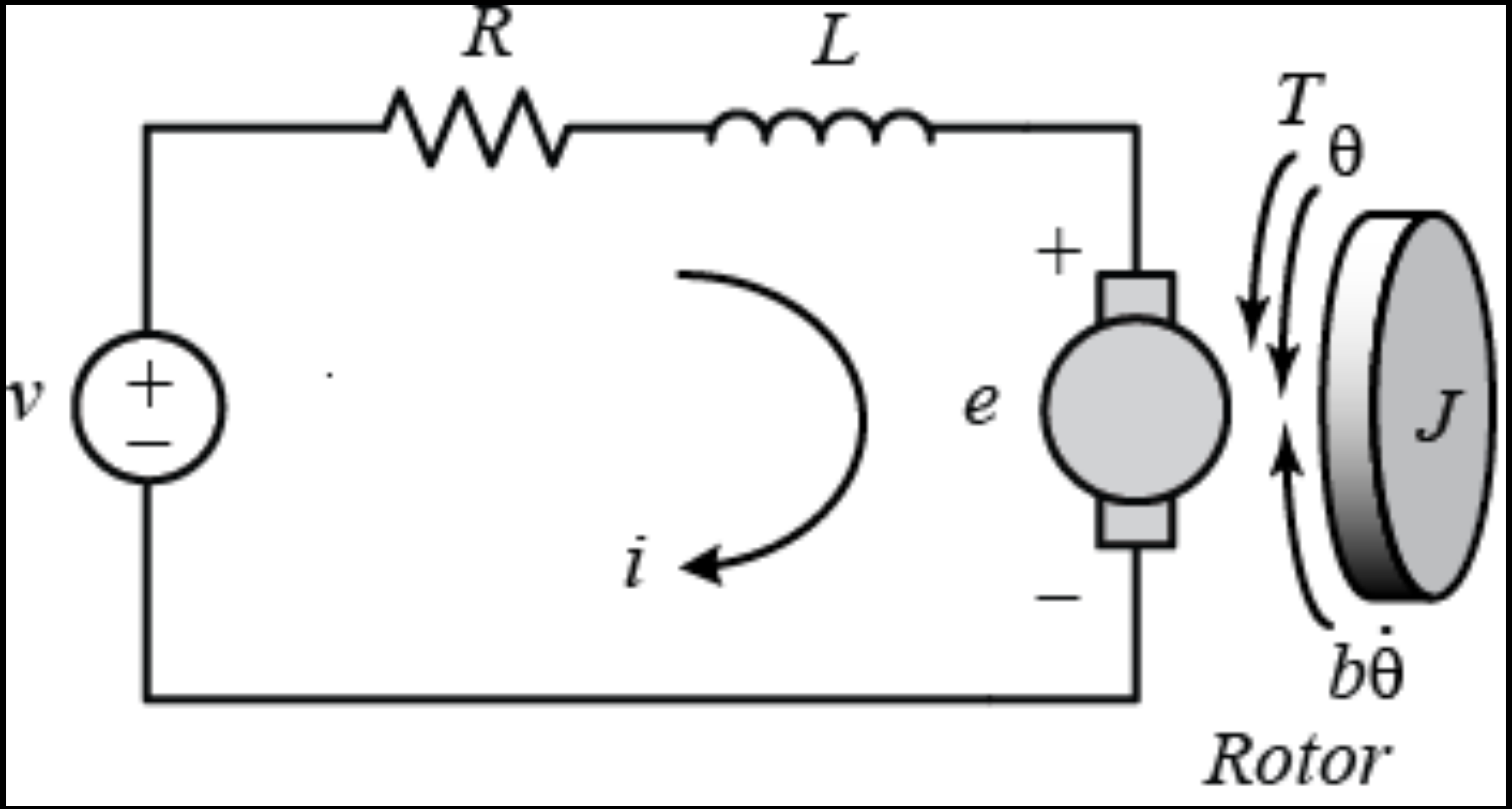
$$R = \frac{L}{2}\frac{w_r + w_l}{w_r - w_l}$$

Interesting observations:

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



Circuit diagram for DC motor



Kirchoff's voltage law

$$L\frac{di}{dt} + Ri = V - e_b$$

Newton's 2nd law

$$J\frac{d\omega}{dt} = \tau - \tau_{load} - b\omega$$

Electro-mechanical coupling

$$e_b = k \cdot \omega$$

$$\tau = k \cdot i$$

J = M.I ~ mass
ω **~ velocity**
τ **~ Force**

Motor control: Magic of closed-loop feedback

Desired properties

1. Need to track reference
2. Should not be dependent on model parameters
3. Provide stable performance (Bounded-Input Bounded-Output)

Adaptive Cruise control

Simpler model problem to track vehicle velocity

Design principles equally apply to EV motor control or anywhere else

Goal: Make a vehicle run at constant reference speed

v_{ref}

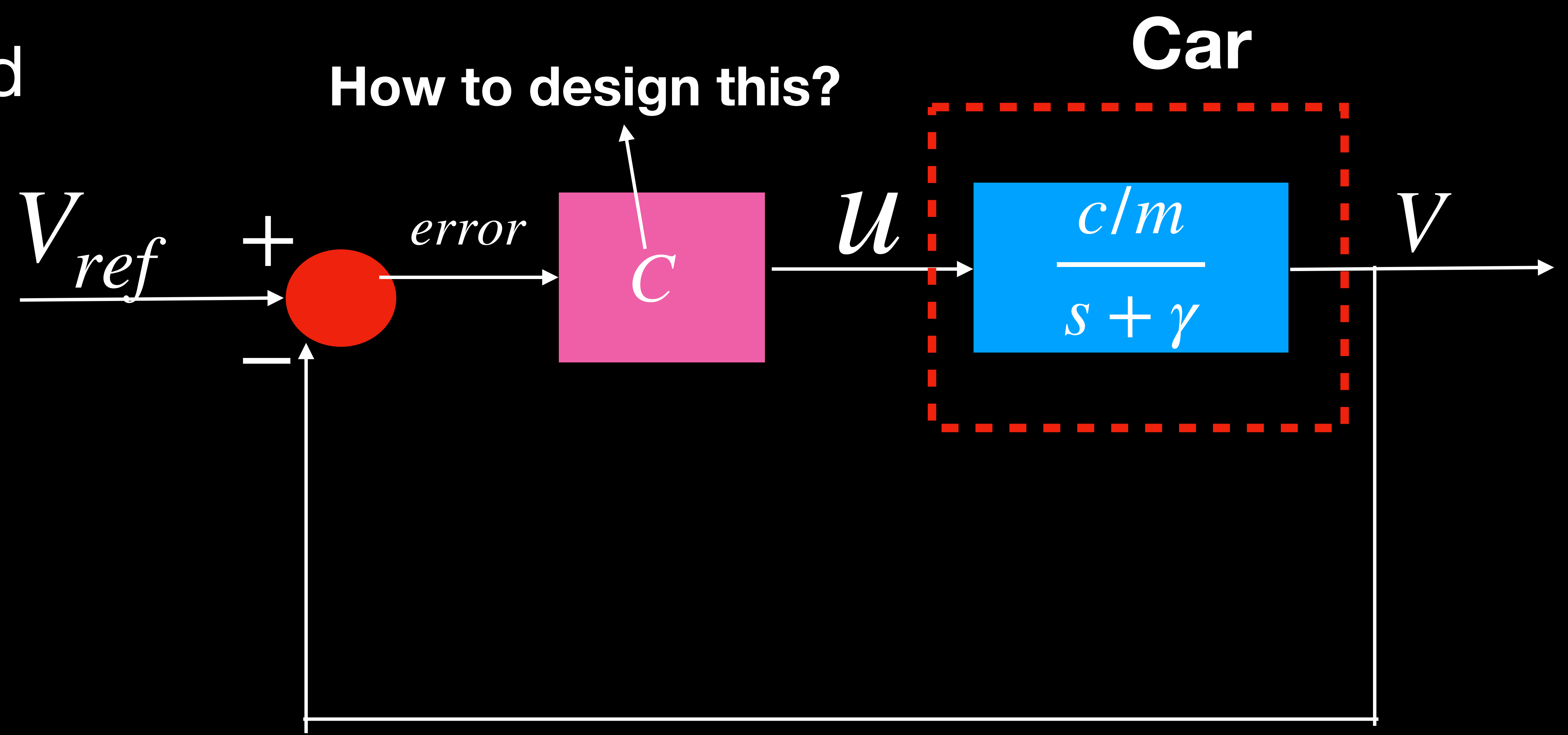
Input: Throttle/ brake (u)

State: Velocity (v)

Newton's 2nd law: $F = ma$ or $a = \frac{1}{m}F$

Assume model: $\dot{v} = \frac{c}{m}u - \gamma v$

- c = vehicle-dependent constant
- m = mass of vehicle
- γ = air-drag coefficient



Example adapted from Control of Mobile robots, Georgia tech

Adaptive Cruise control design

Three design choices

1. Bang-Bang control

If error is positive, hit max throttle or else hit max brake

$$\dot{v} = \frac{c}{m}k(v_{ref} - v) - \gamma v$$

2. Proportional control

Control is proportional to error

Steady-state $\implies \dot{v} = 0$

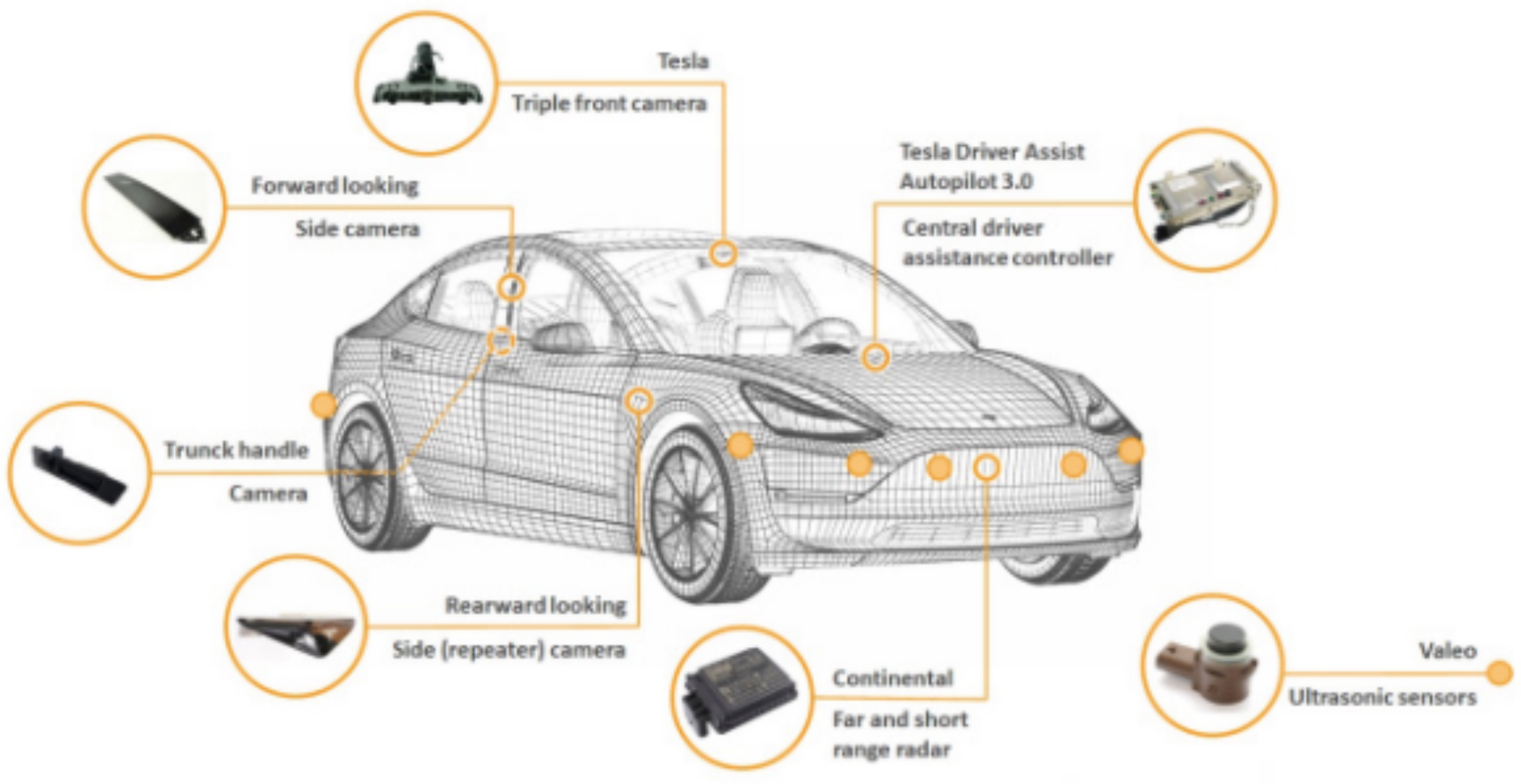
$$v_{ss} = \frac{ck}{ck + m\gamma}v_{ref}$$

3. PI control

Control is proportional to both current error as well as accumulation of previous error terms

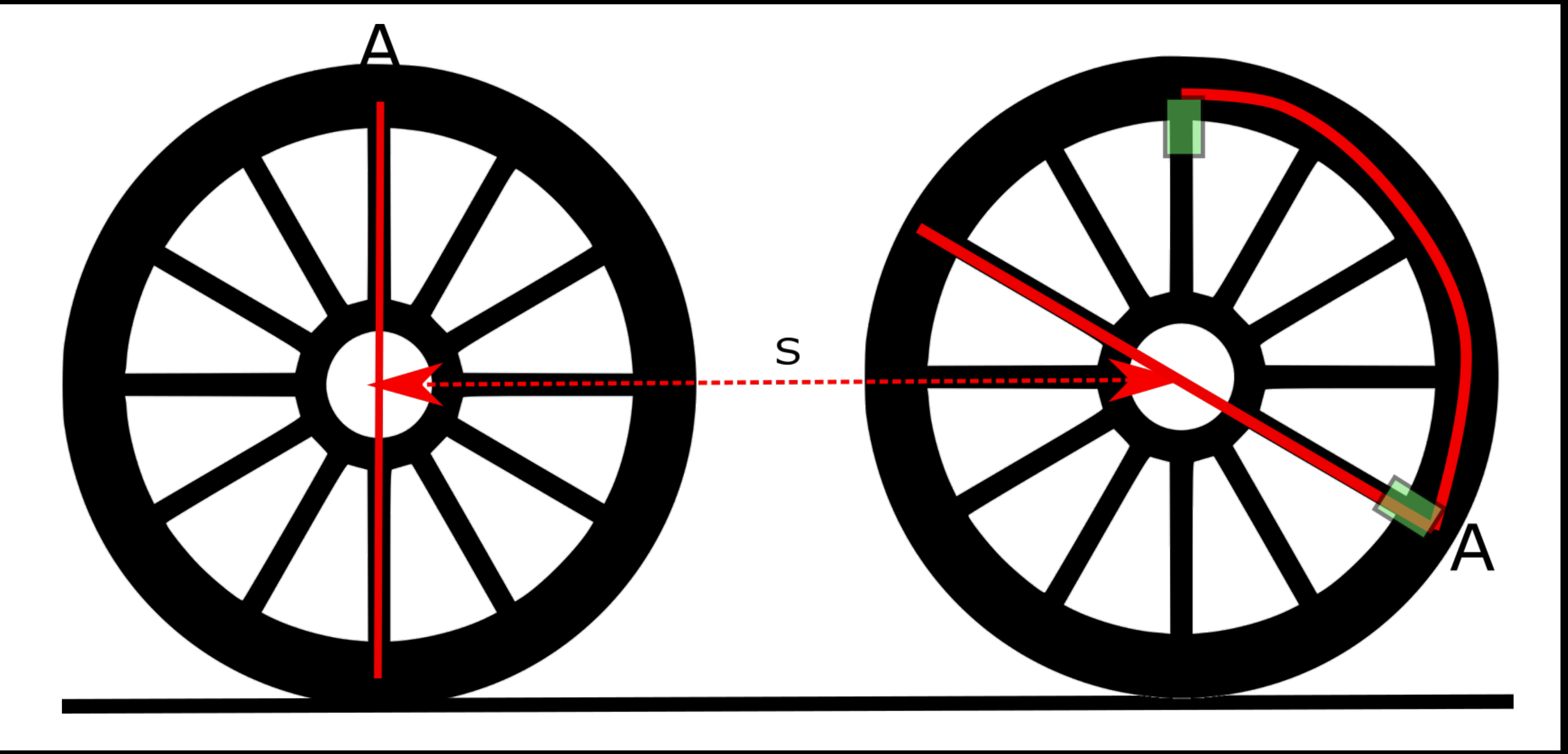
Illustrations in Jupiter notebook

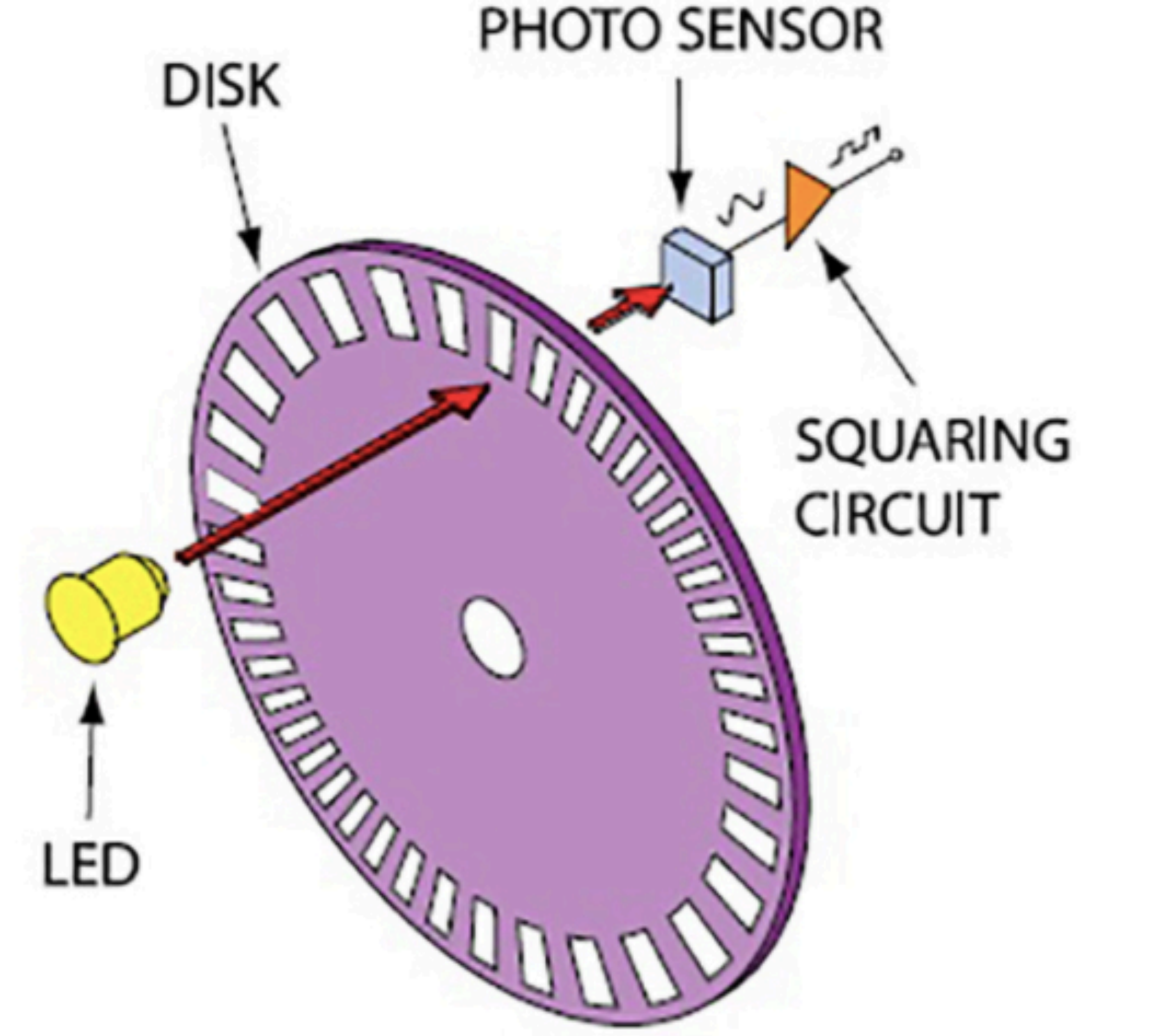
Sensor placements in Model 3

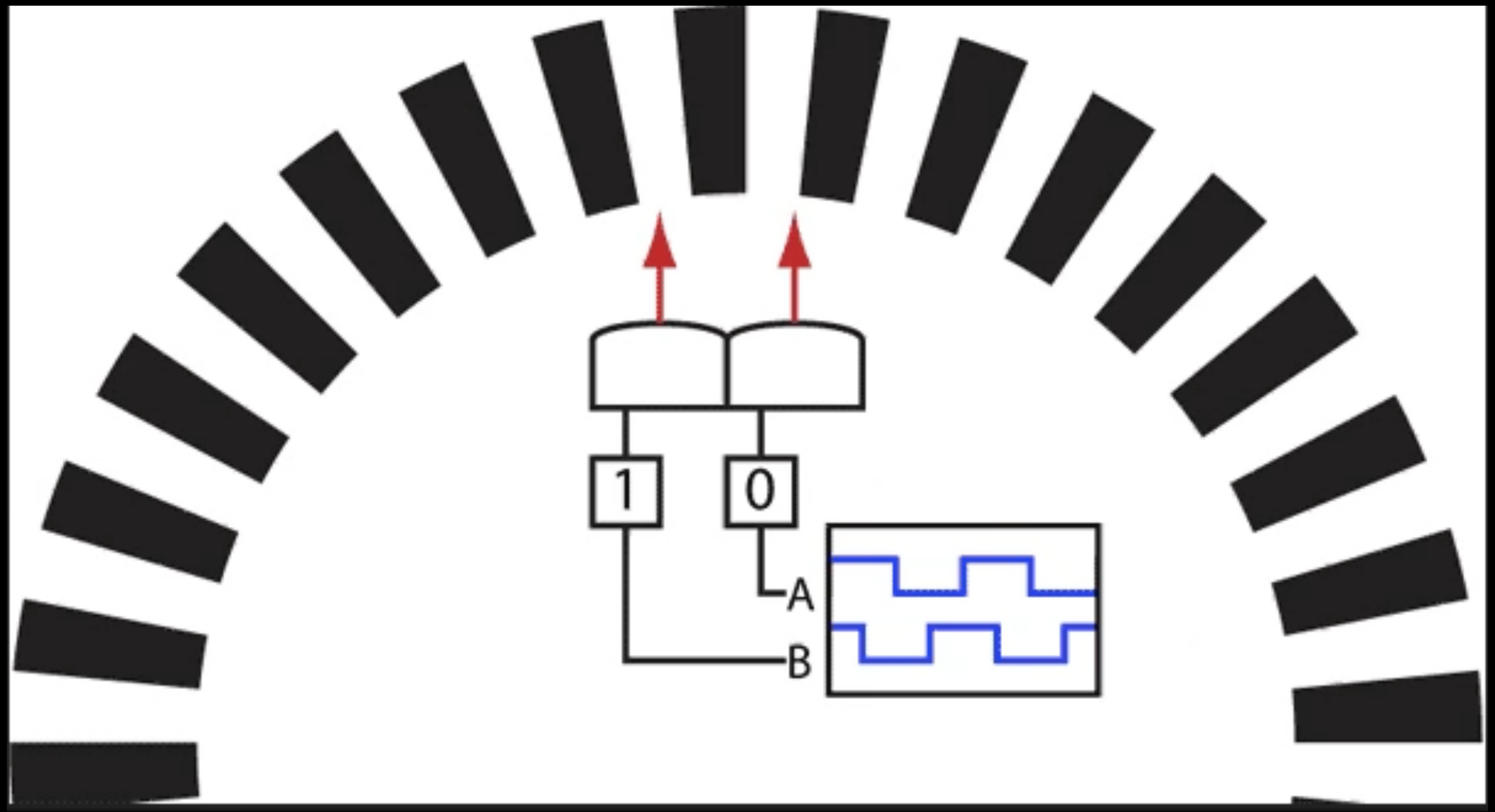


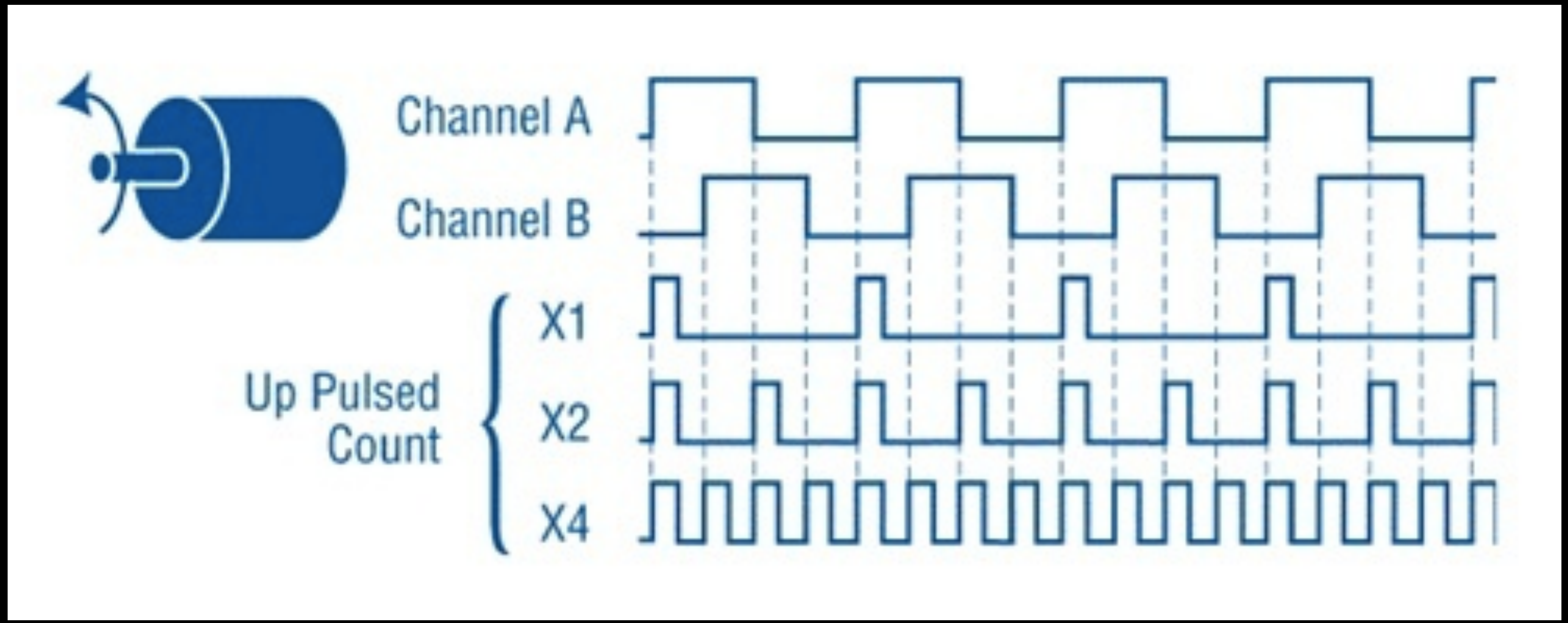
Wheel encoders

Source: Anaheim Automation









Wheel encoders

Every T ms

Encoder reports # ticks wheel moved = n

Encoder resolution = N

N ticks complete 1 revolution

Distance moved = $2\pi R \frac{n}{N}$

Incremental vs Absolute encoders

Absolute encoder: Multiple concentric, perforated circles

When would we need an absolute encoder?

Think position versus velocity

Wheel encoders

Usage

- Smallest movement that can be measured is $\frac{2\pi R}{N}$

- Largest speed that can be measured is $\frac{2\pi R}{dt}$

Pros:

- Fairly accurate estimates of linear/ angular velocity
- Distances and rotations are accurate in short-term

Cons:

- Vehicle position "drifts" when v, ω is integrated over longer periods. Encoder noise gets accumulated

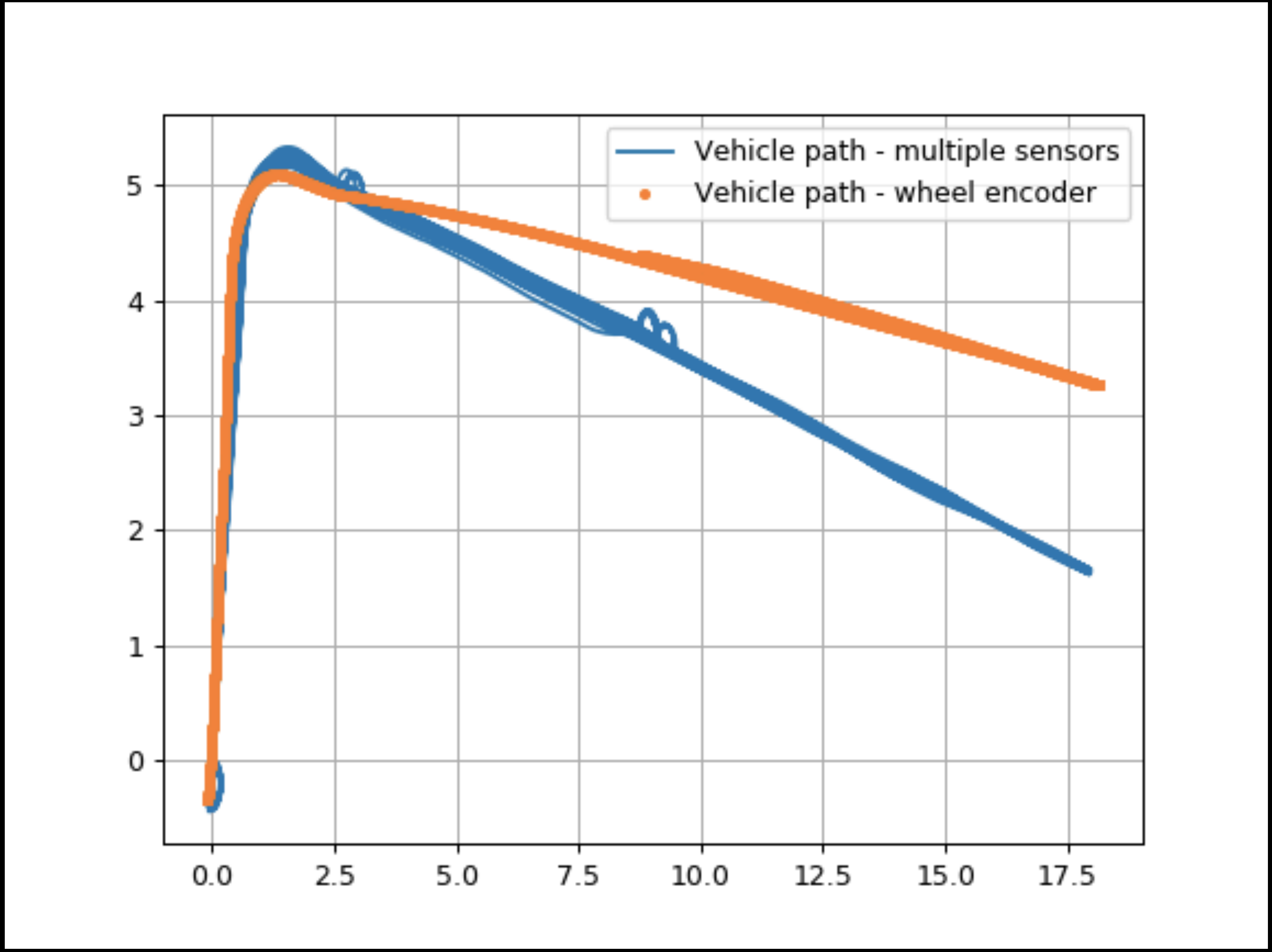


Illustration in Jupiter notebook

Lidar

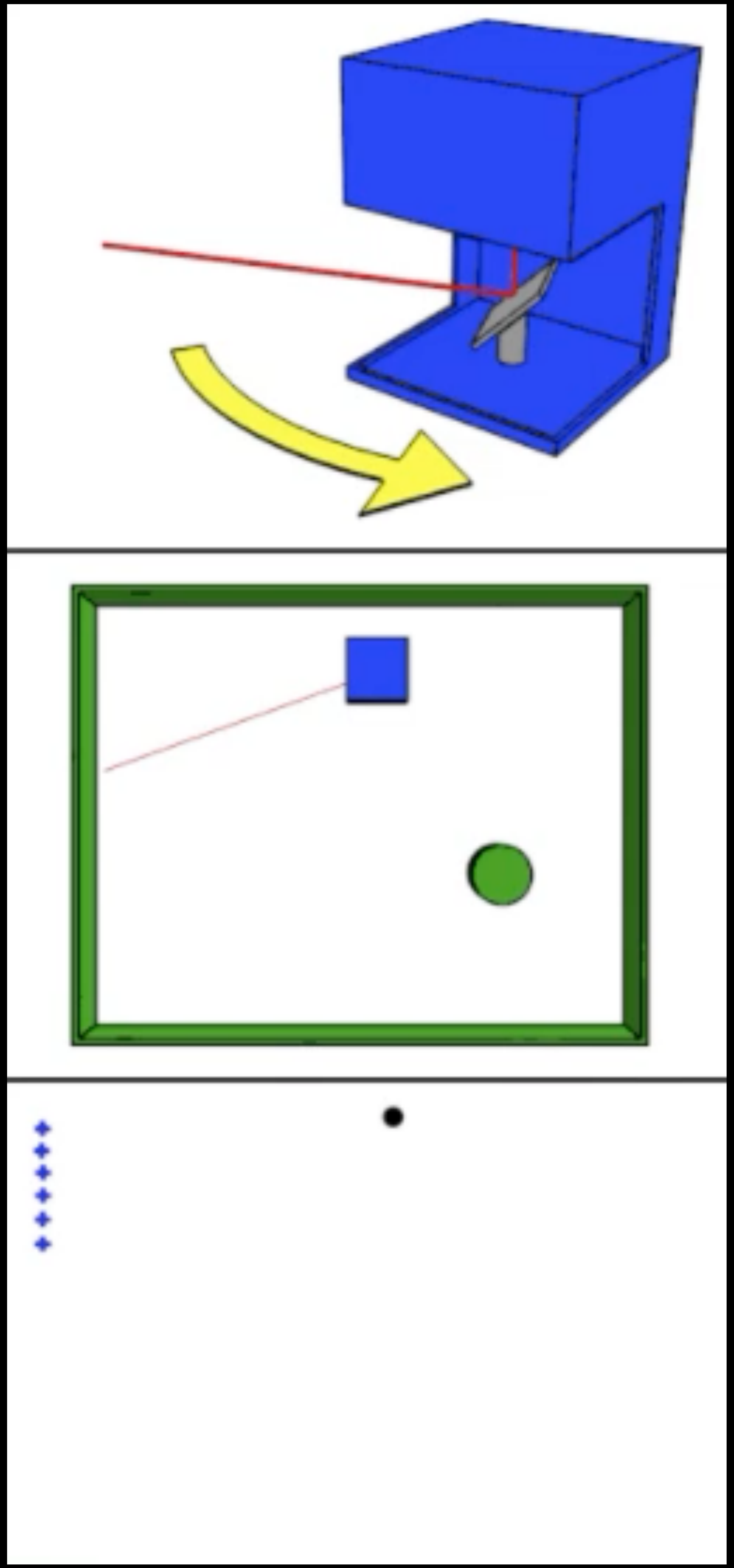
Light detection and ranging

Visible or Near-IR light used to image objects

Object distance calculated from time-of-flight estimates

Several advantages

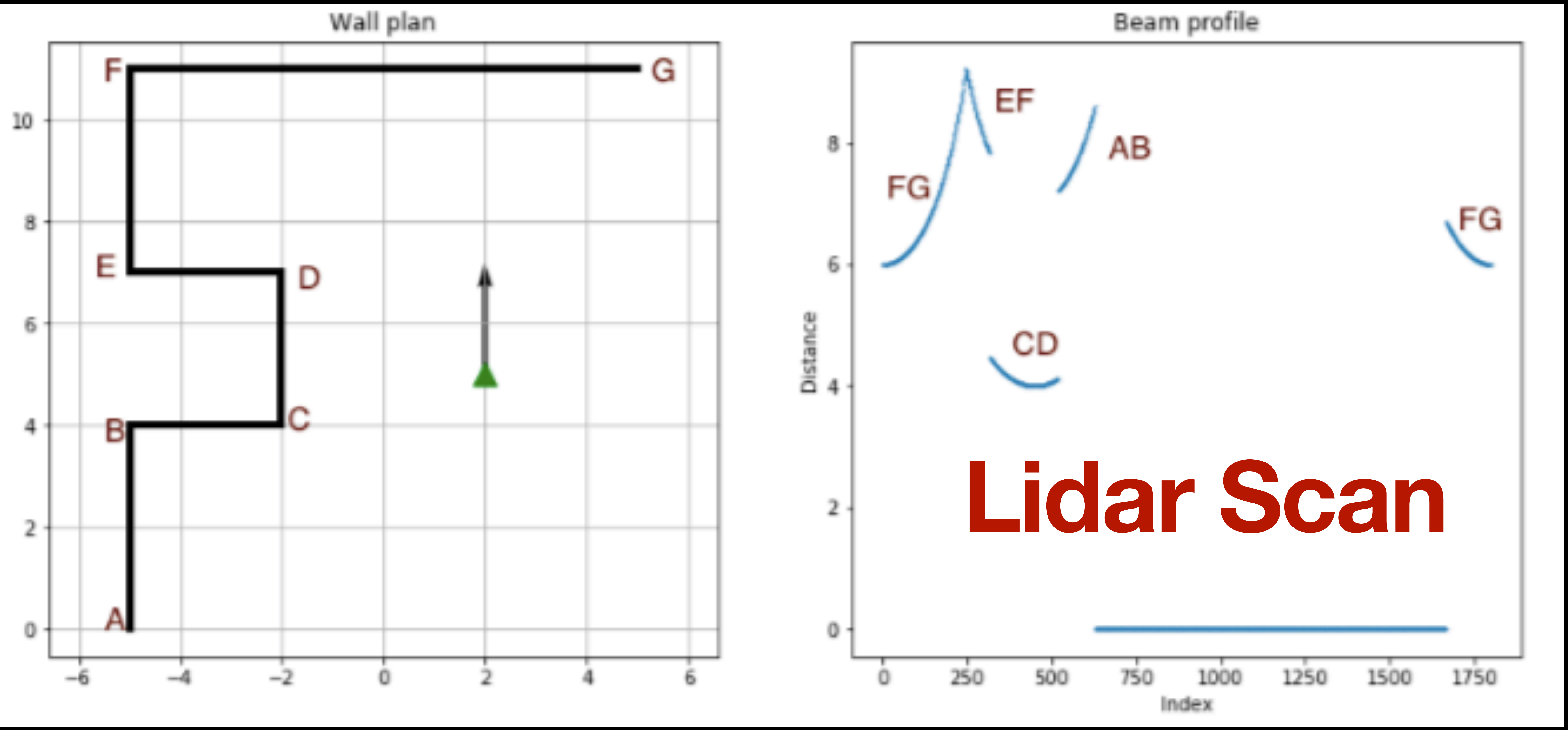
- Low Noise floor
- Independent of ambient lighting
- “Ready-made” 3-D imaging

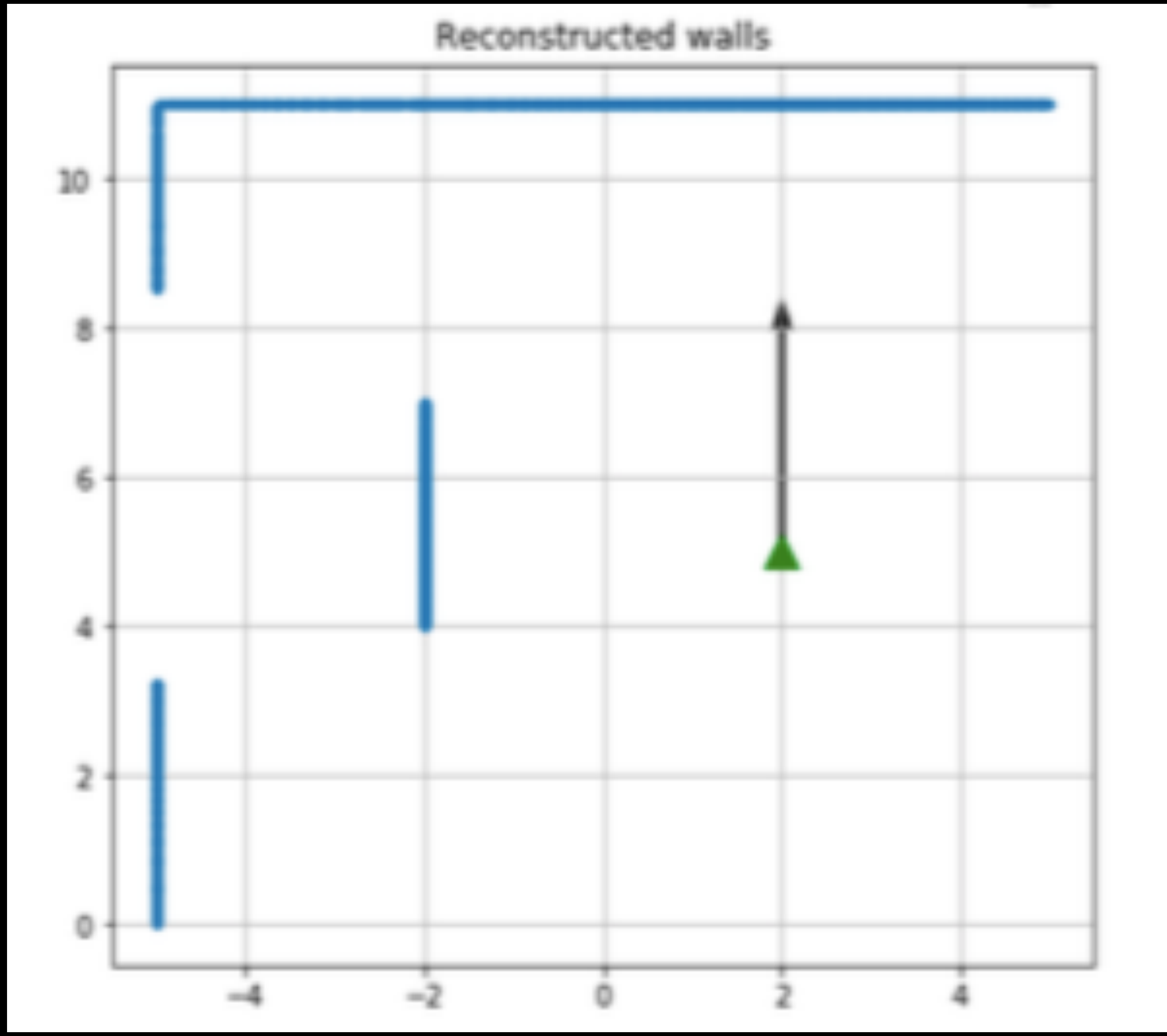


Video: https://en.wikipedia.org/wiki/Lidar

2-D Lidar

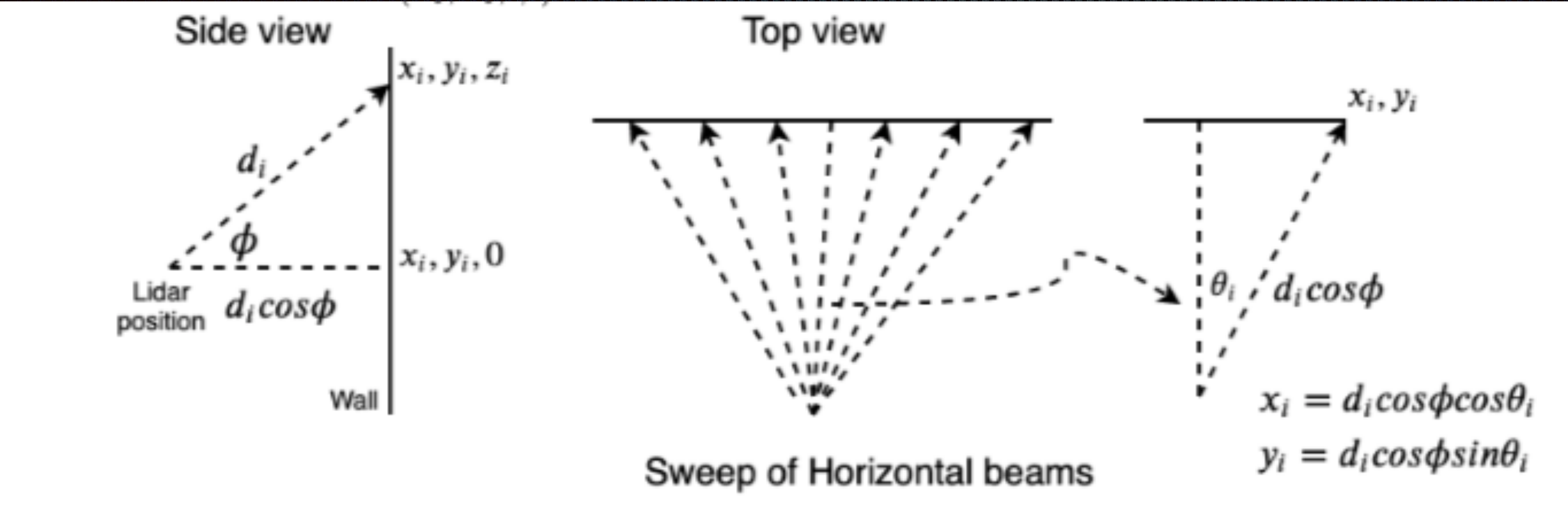
Single Laser beam rotating 360 degrees
Return reflections d recorded along with azimuth angle θ





2-D Lidar simulator animation in Jupiter notebook

3-D Lidar



N-laser beams rotating 360 degrees
Span different elevation angles ϕ

Object which reflects is at (d, θ, ϕ)

- **(θ, ϕ) is given by beam position**
- **d is calculated by time-of-flight**

Can be converted to cartesian coordinates (x, y, z)

3-D Lidar visualization