# Simple Path Following for Differential Drive Robots

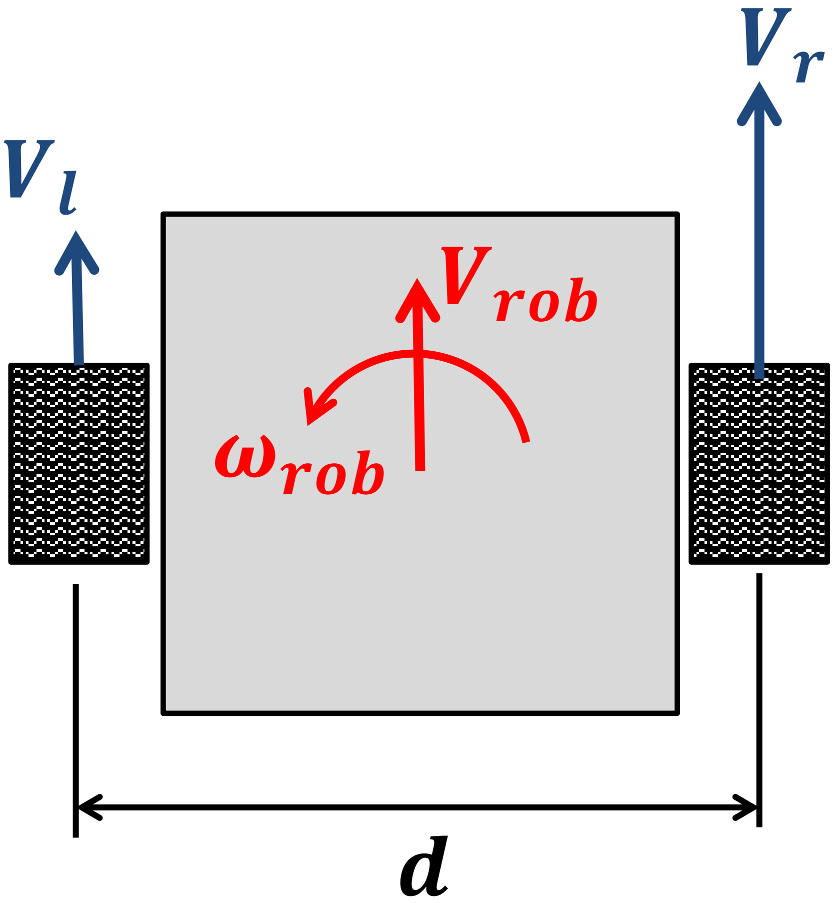
Nikolai Kummer

Companion Slides for

**03: Path Planning with a Differential Drive Robot | V-Rep Tutorial**

**https://www.youtube.com/watch?v=OfpB87pRoUk**

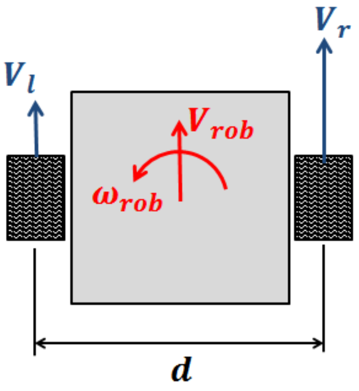
# Differential Drive Kinematics



# Differential Drive Velocity

- Velocity:

$$V_{rob} = \frac{V_r + V_l}{2}$$



- Rotational Speed:

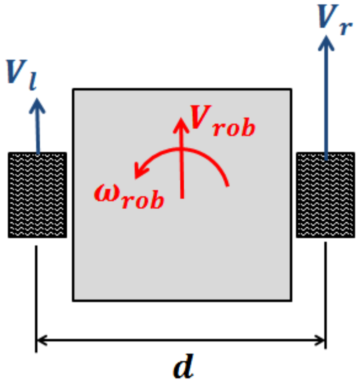
$$\omega_{rob} = \frac{V_r - V_l}{d}$$

# Differential Drive Velocity

- Solve for $V_r, V_l$:

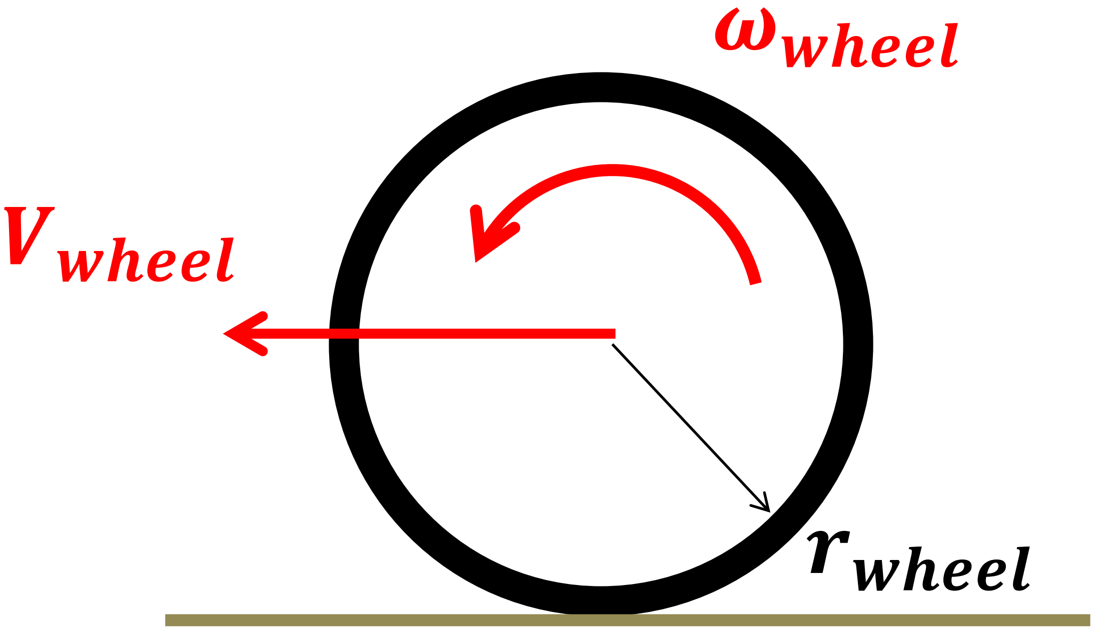
$$V_r = V_{rob} + \frac{d}{2}\omega_{rob}$$

$$V_l = V_{rob} - \frac{d}{2}\omega_{rob}$$



# Wheel Rotation Velocity Relationship:

$$V_{wheel} = \omega_{wheel} r_{wheel}$$



- Assume no slipping

# Desired velocity: v_des
# Desired Rotation rate: om_des

```
--Lua Code
v_des=0.1
om_des=0.1
d=0.06          --wheels separation

v_r=(v_des+d*om_des)
v_l=(v_des-d*om_des)
```

Desired velocity: v_des

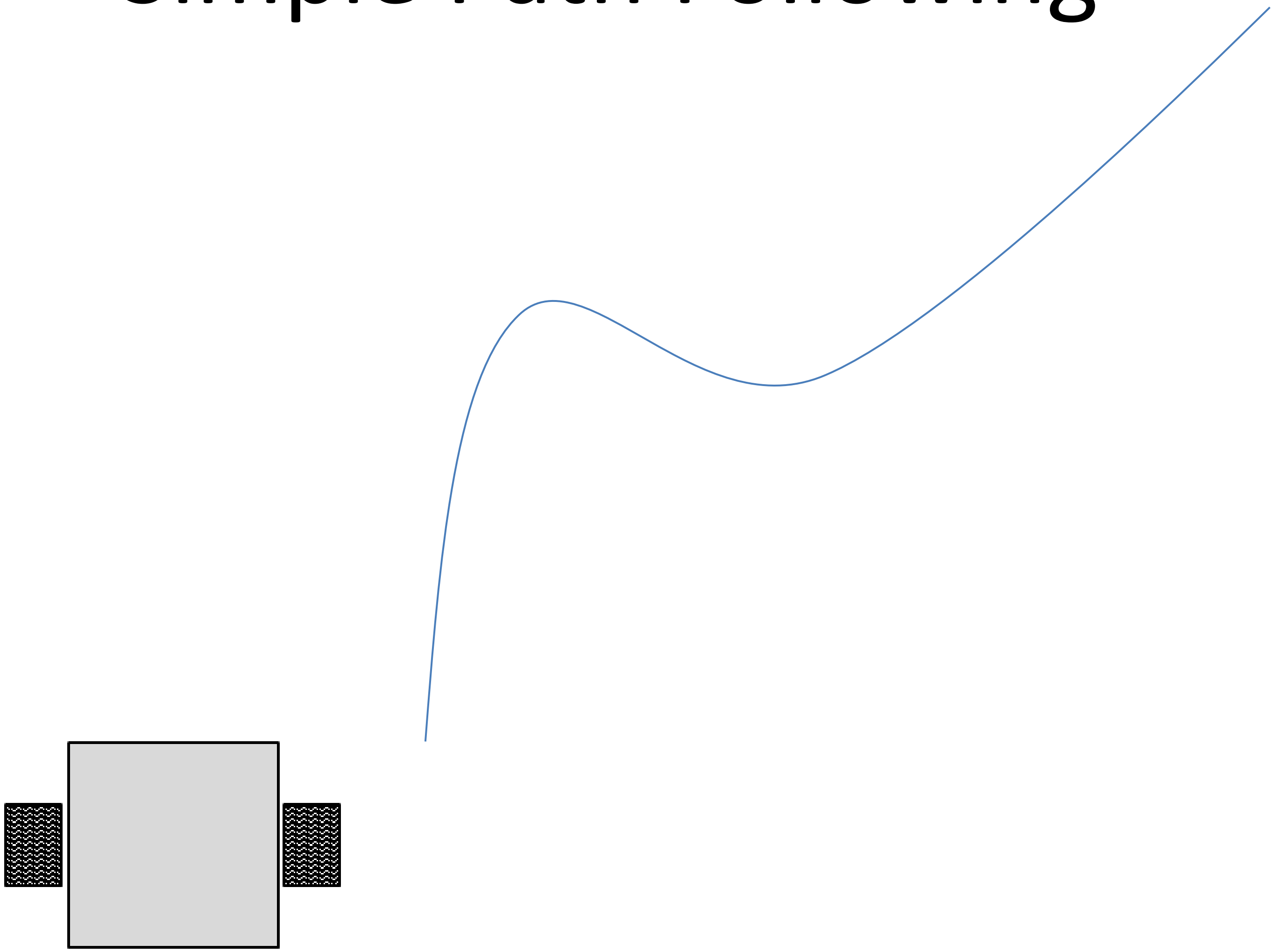
Desired Rotation rate: om_des

```
--Lua Code

r_w=0.0275        --wheel radius
omega_right=v_r/r_w
omega_left=v_l/r_w

simSetJointTargetVelocity(rm,-omega_right)
simSetJointTargetVelocity(lm,-omega_left)
```

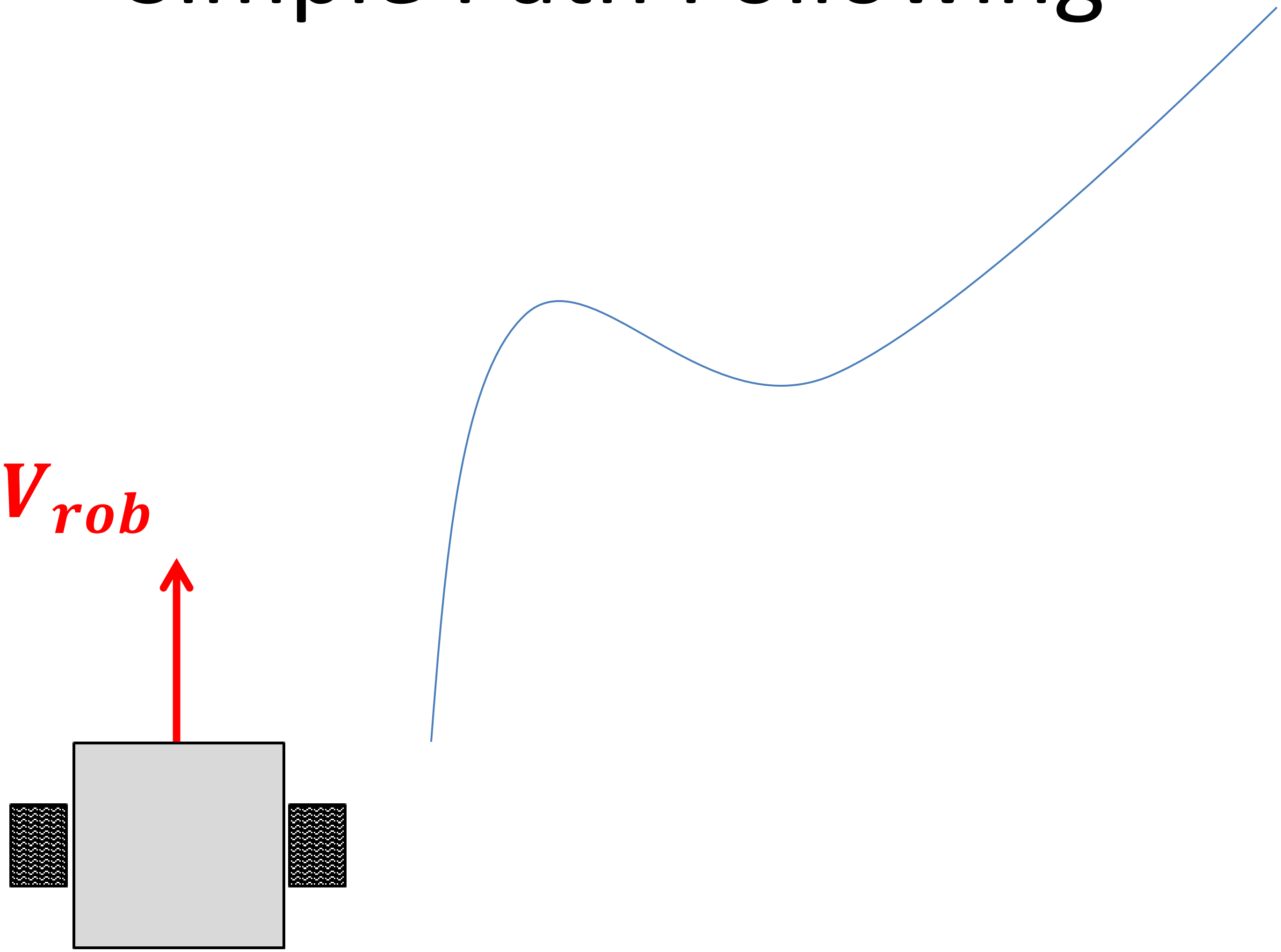
# Control Path-Planning Dialog from Lua

```
--Get handle to Path planning task
path_plan_handle=simGetPathPlanningHandle('PathPlanningTask')

--Search Path
planstate=simSearchPath(path_plan_handle,5)
--5 = max search time
```

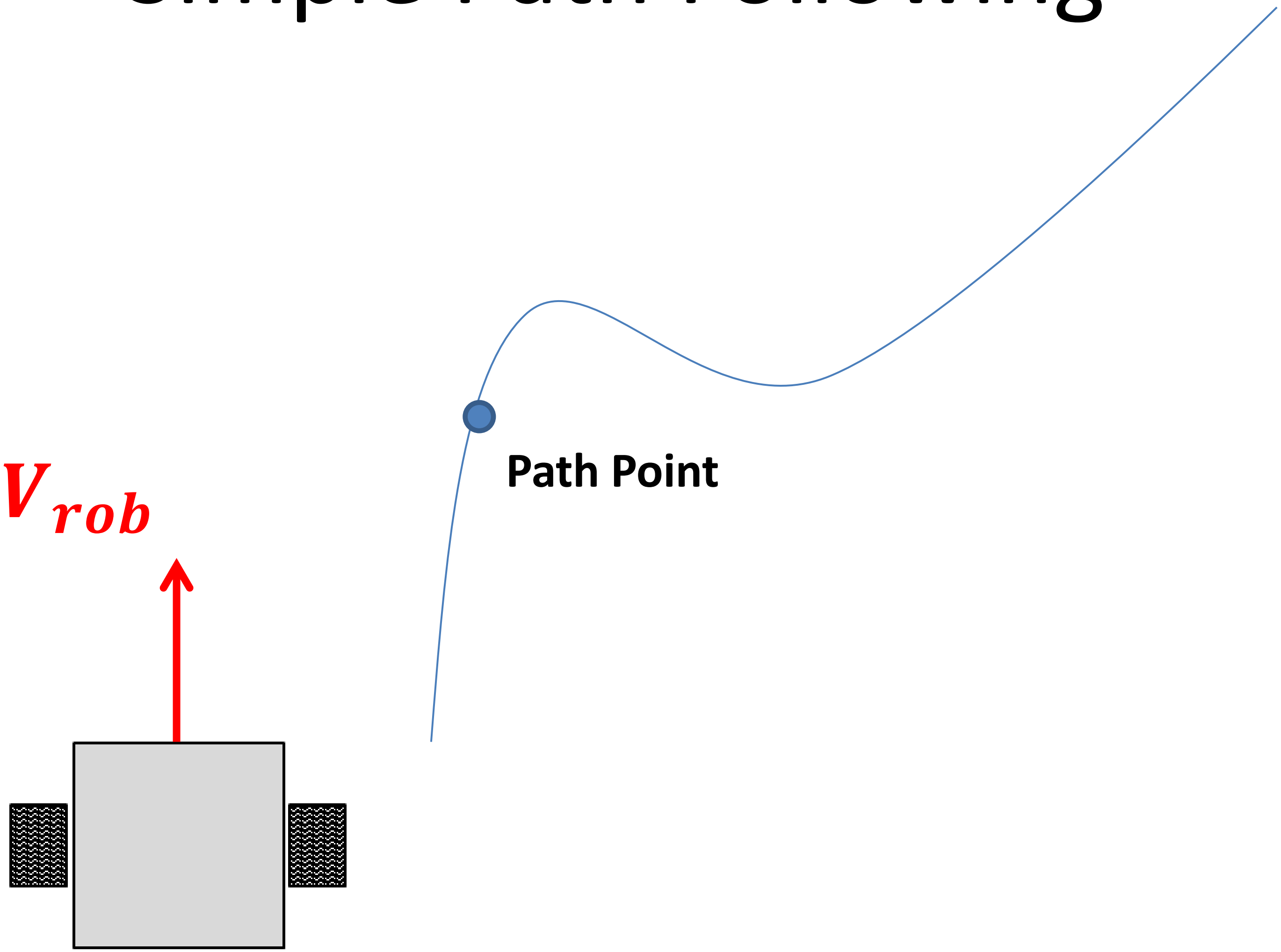
# Simple Path Following



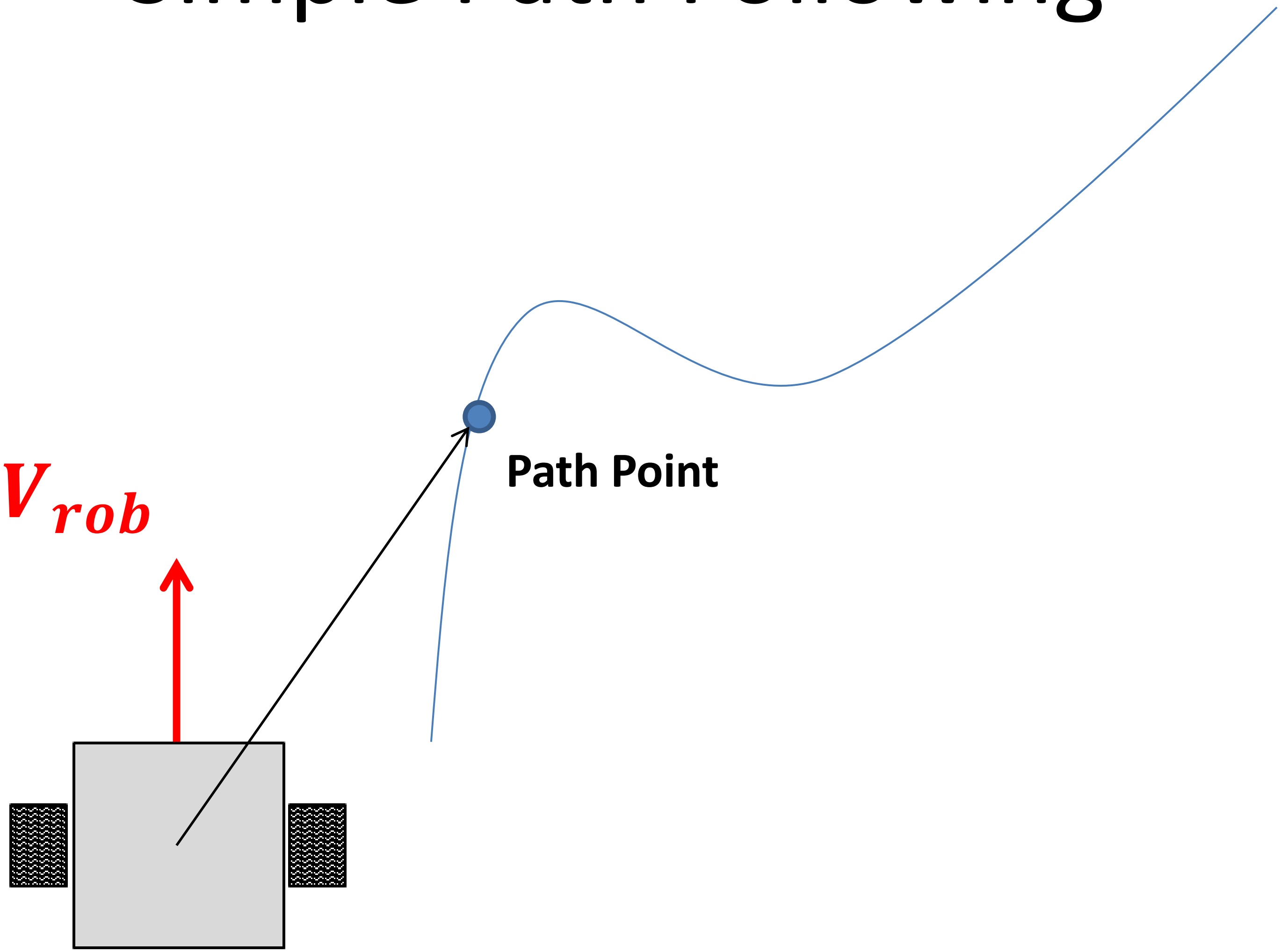
# Simple Path Following



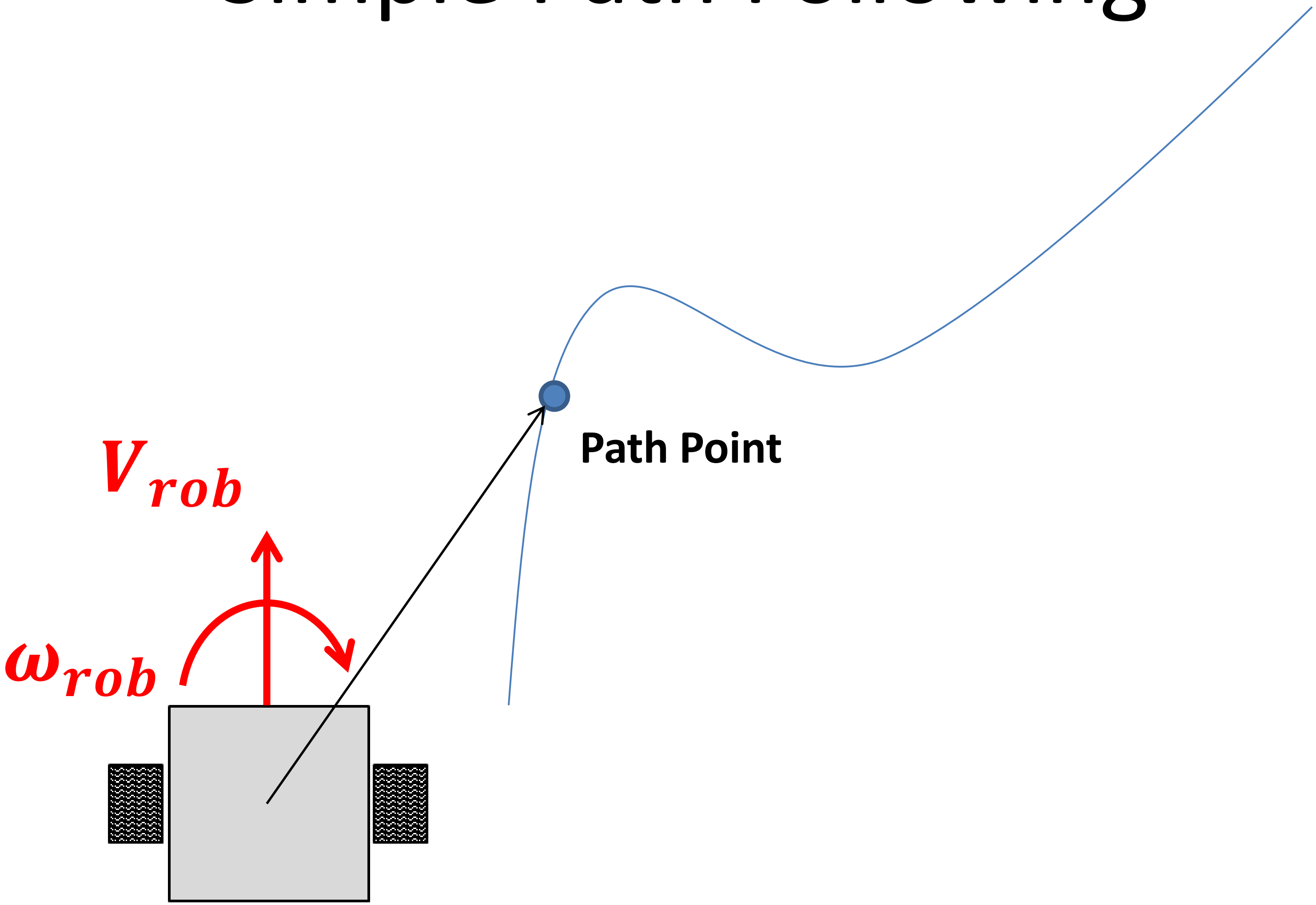
# Simple Path Following



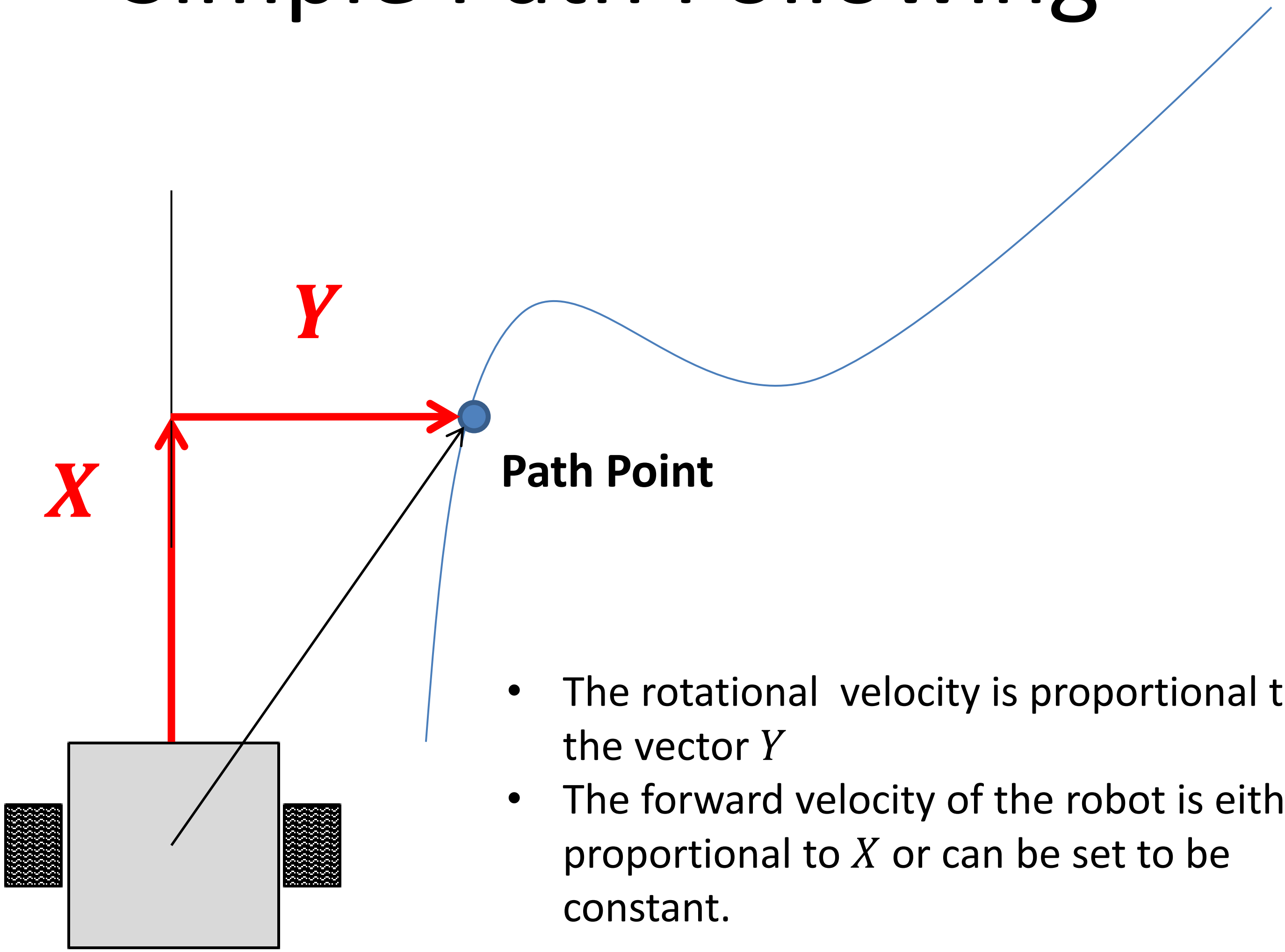
# Simple Path Following



# Simple Path Following

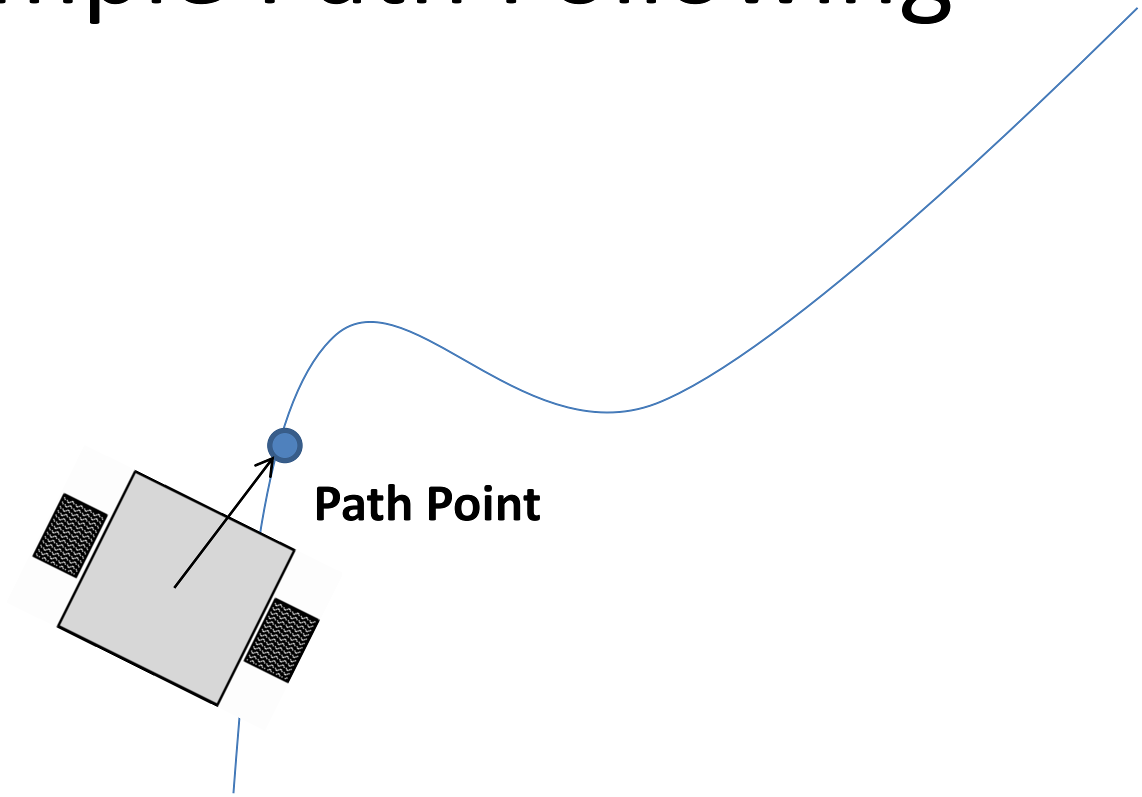


# Simple Path Following

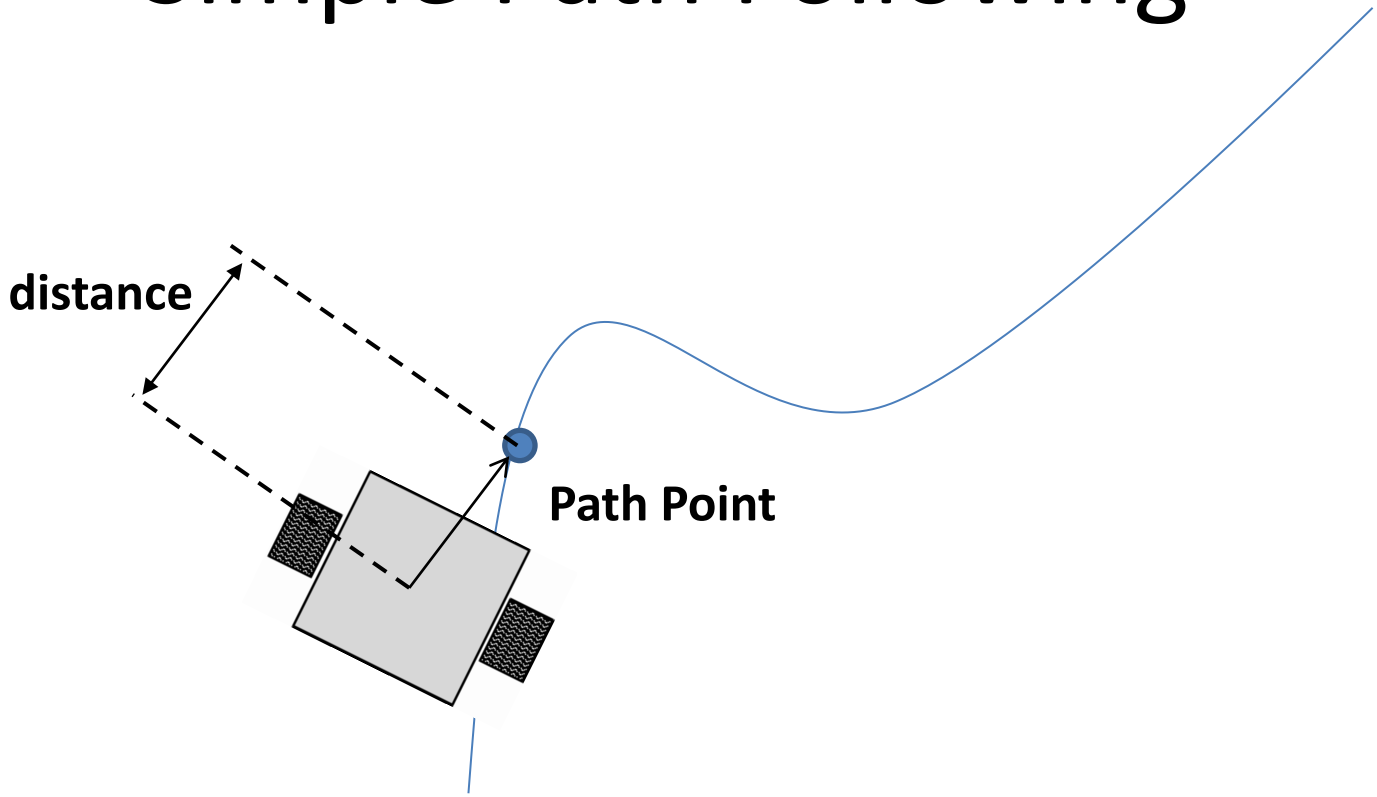


- The rotational velocity is proportional to the vector $Y$
- The forward velocity of the robot is either proportional to $X$ or can be set to be constant.

# Simple Path Following

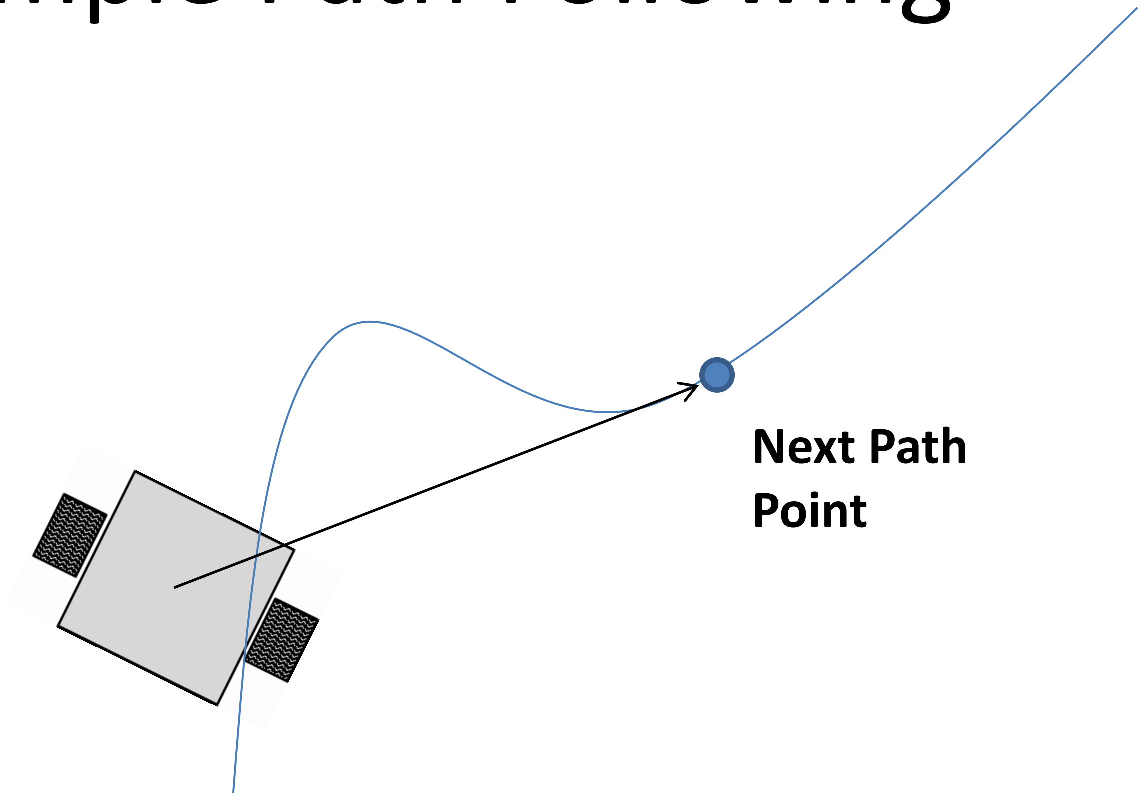


# Simple Path Following



When the distance is less than a specified limit, the path point advances

# Simple Path Following


Next Path Point