FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
department of control and instrumentation

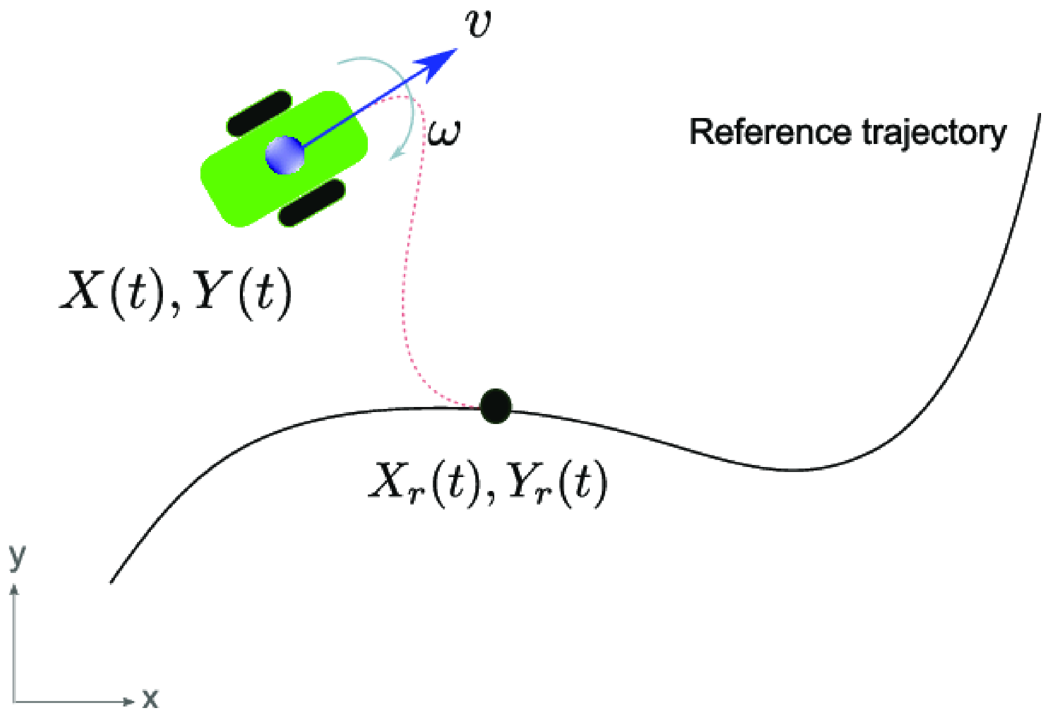
3 – Kinematics and Motion control

Advanced Methods for Mapping and Self-localization in Robotics
MPC-MAP

Tomas Lazna
Brno University of Technology
2024

What is motion control for?

- Following known trajectory
- Kinematic model is required –
  a relation between the wheels speed
  and the robot motion
- Are there any constraints?
- How to model motion?



[1]

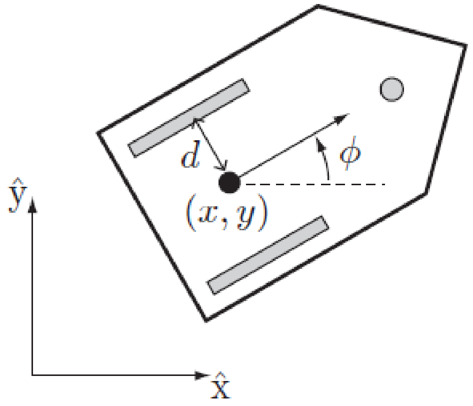
1. ABDELWAHAB, Mohamed et al. Trajectory Tracking of Wheeled Mobile Robots Using Z-Number Based Fuzzy Logic. *IEEE Access* [online]. 2020, **8**, 18426-18441 [cit. 2022-02-10]. DOI: 10.1109/ACCESS.2020.2968421

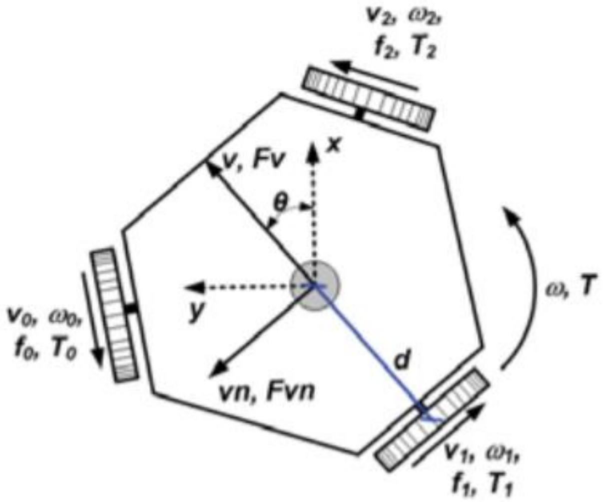
Kinematics

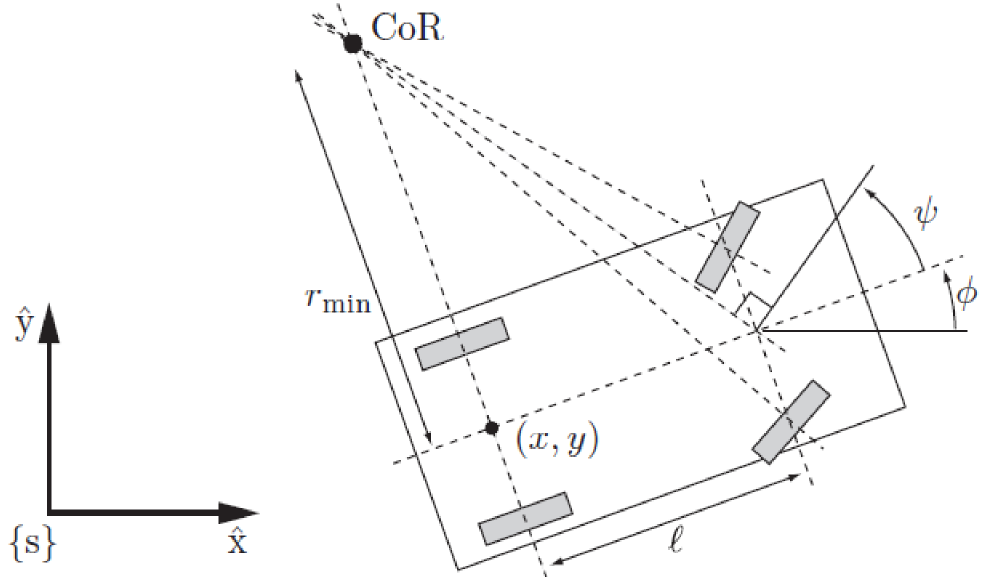
Describing and modeling the robot movement

Introduction

- Types of drive
  - Differential
  - Ackermann
  - Omnidirectional
  - Others …
- Kinematics
  - „Geometry of motion“
  - Relation (acceleration –) velocity – position
  - Cause of motion is not analyzed
- Dynamics
  - Focuses on „why is object moving“
  - Forces, torques, mass, etc.





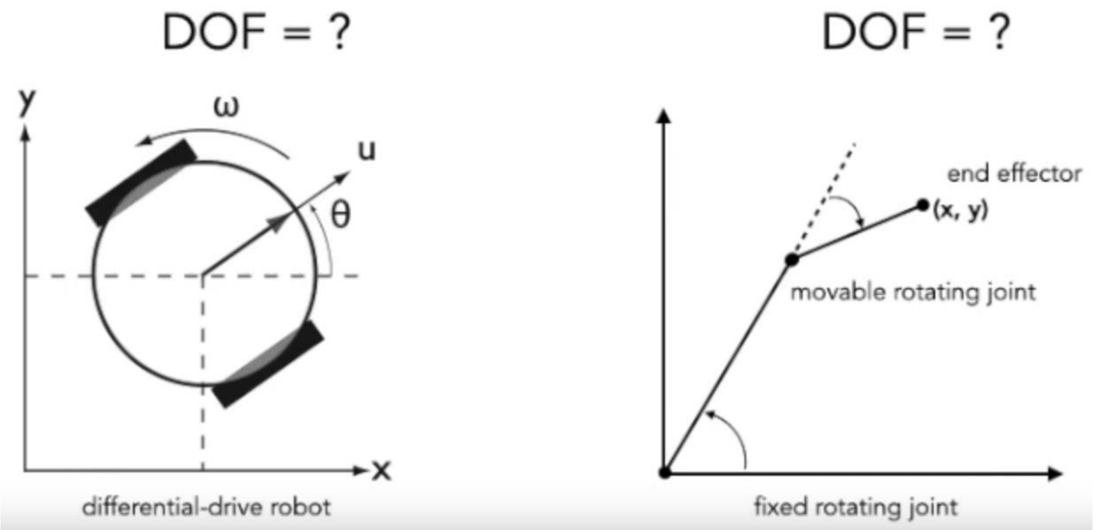


[1]

1. Electronics and Robotics Club. The ERC Handbook: Drive Mechanism [online]. BITS Goa, 2020 [cit. 2022-02-14]. Available at: https://erc-bpgc.github.io/handbook/mechanical/drive_mechanism/

Holonomic vs. non-holonomic

- Degrees of freedom (DOFs)
  - Minimum number of real numbers to represent the robot‘s configuration
  - Most actuators control a single DOF
  - Either translational or rotational
  - Depends on the type of robot
- Degrees of motion (DOMs)
  - = Differentiable DOFs (DDOFs)
  - Number of DOFs that can be directly accessed by the actuators
  - Also number of independent motion velocities
    - Unicycle, Bicycle
    - = 2 = 1



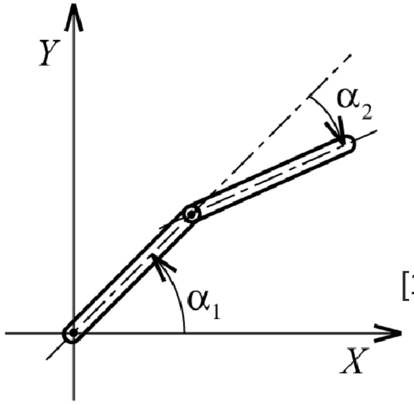
[1]

1. PROROK, Amanda. Mobile Robot Systems, Lecture 3: Robot Motion & Control. In: *YouTube* [online]. 2021 [cit. 2022-02-16]. Available at: https://youtu.be/tocxIukOa-I

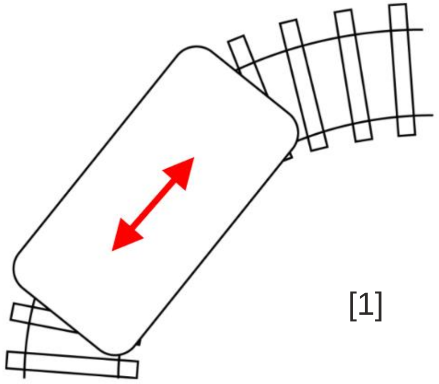
Holonomic vs. non-holonomic

- Constraints
  - Constrain position = holonomic
  - Constrain velocity = non-holonomic
- Integrability of constraints
  - $f(\boldsymbol{q}, t)$ – depends only on position = integrable = holonomic
  - $f(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ – depends also on velocity = is not integrable = non-holonomic
- Holonomic robot
  - DOF = DOM (= DDOF)
  - All constraints are holonomic
- Non-holonomic robot
  - DOF > DOM
  - Position depends on the order of control inputs!

Examples of holonomic systems:



[1]

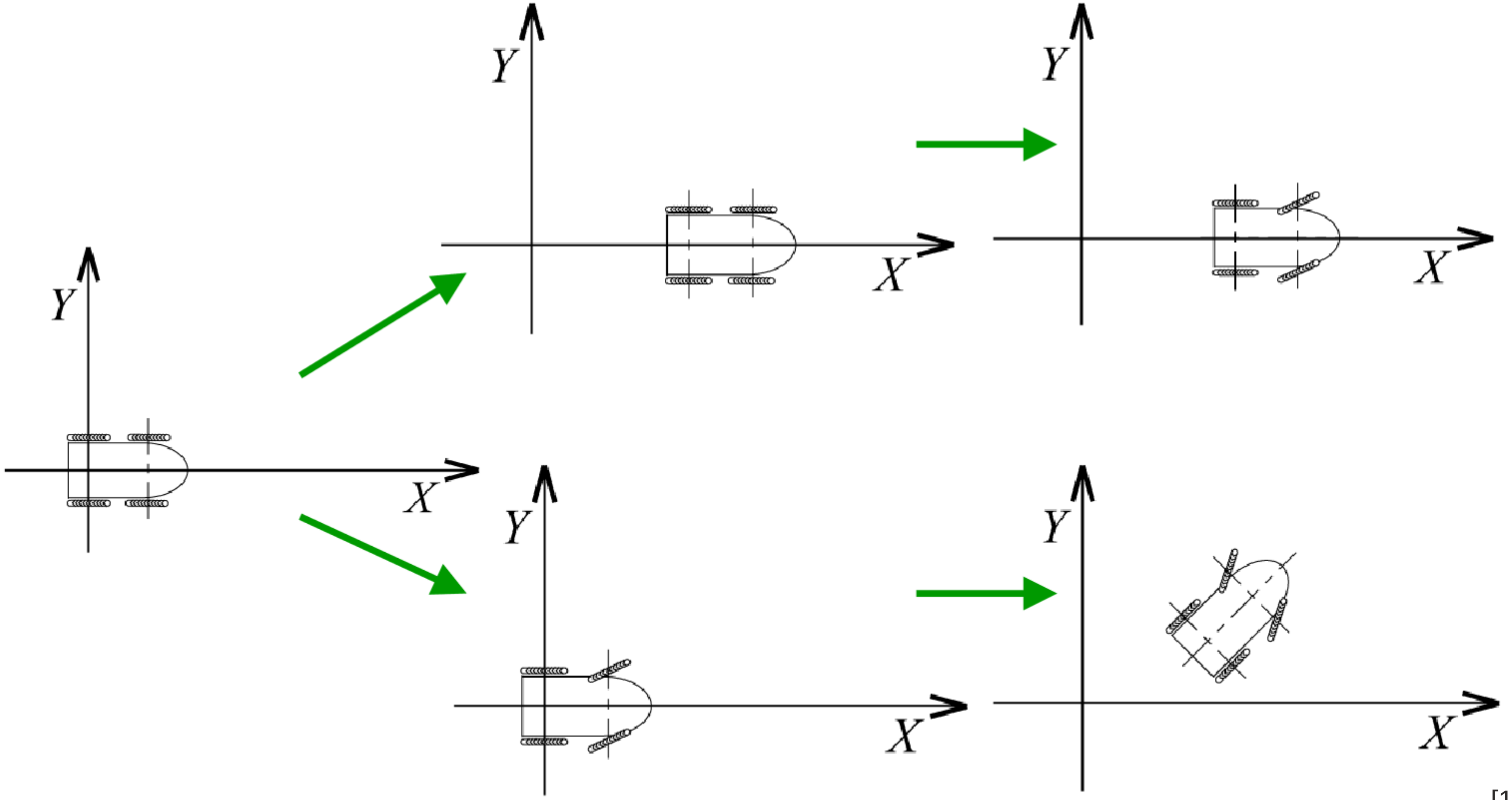


[1]

[2]

1. HRABEC, Jakub. Kinematika mobilních robotů: Stručný úvod do problematiky. Prezentace pro předmět MPC-RBT. Vysoké učení technické v Brně, 2021.
2. Robotics Lab [online]. Nairi-Tech, 2014 [cit. 2021-02-28]. Available at: http://nairi-tech.com/products/view/robotics-lab.html

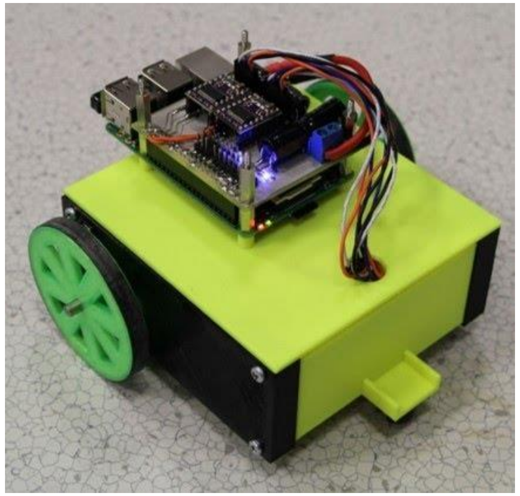
Holonomic vs. non-holonomic



[1]

Depends on the order!

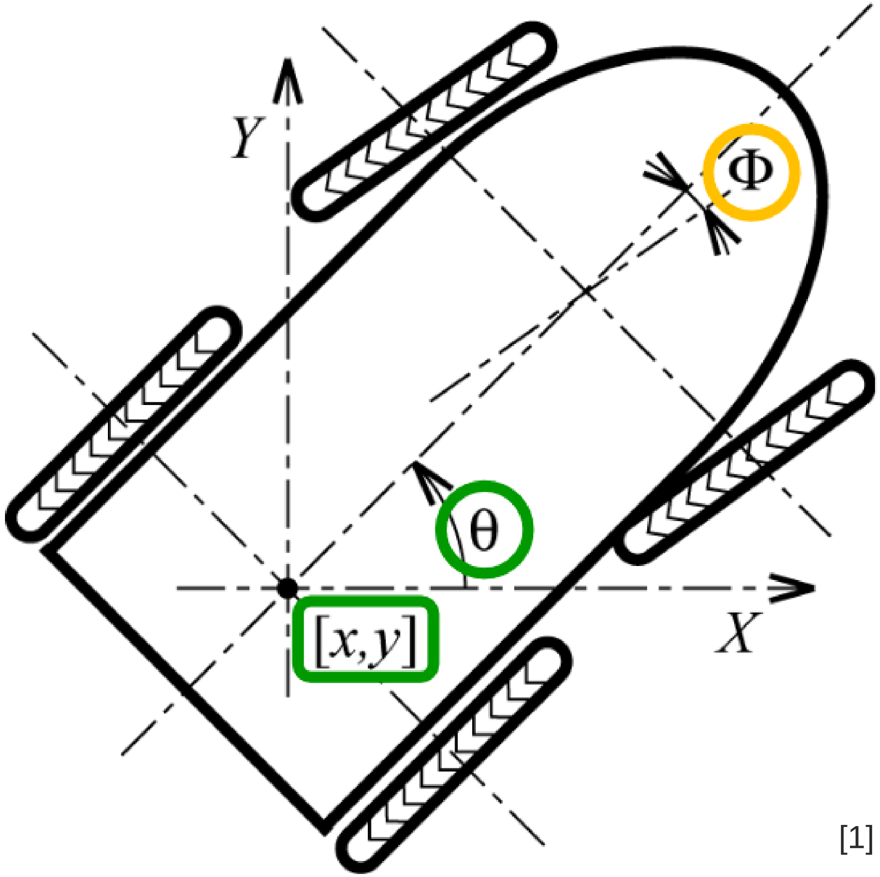
Differential drive = non-holonomic



1. HRABEC, Jakub. Kinematika mobilních robotů: Stručný úvod do problematiky. Prezentace pro předmět MPC-RBT. Vysoké učení technické v Brně, 2021.

State of the robot in 2D

- State
  - Position of the reference point in the world frame – x, y
  - Orientation – θ
  - Describes configuration of the robot body
  - $\boldsymbol{x} = (x, y, \theta)$
- Extended state
  - State + additional parameters
  - E. g., steering angle Φ in Ackermann drive
  - $\boldsymbol{x}_{\mathrm{Ack}} = (x, y, \theta, \Phi)$



[1]

1. HRABEC, Jakub. Kinematika mobilních robotů: Stručný úvod do problematiky. Prezentace pro předmět MPC-RBT. Vysoké učení technické v Brně, 2021.

Differential drive

- Parameters of the chassis
  - Wheel radius r
  - Distance between wheels d
- Control inputs
  - Speed of wheels ω_R, ω_L

- Simplification to unicycle
  - Forward velocity v
  - Angular velocity ω

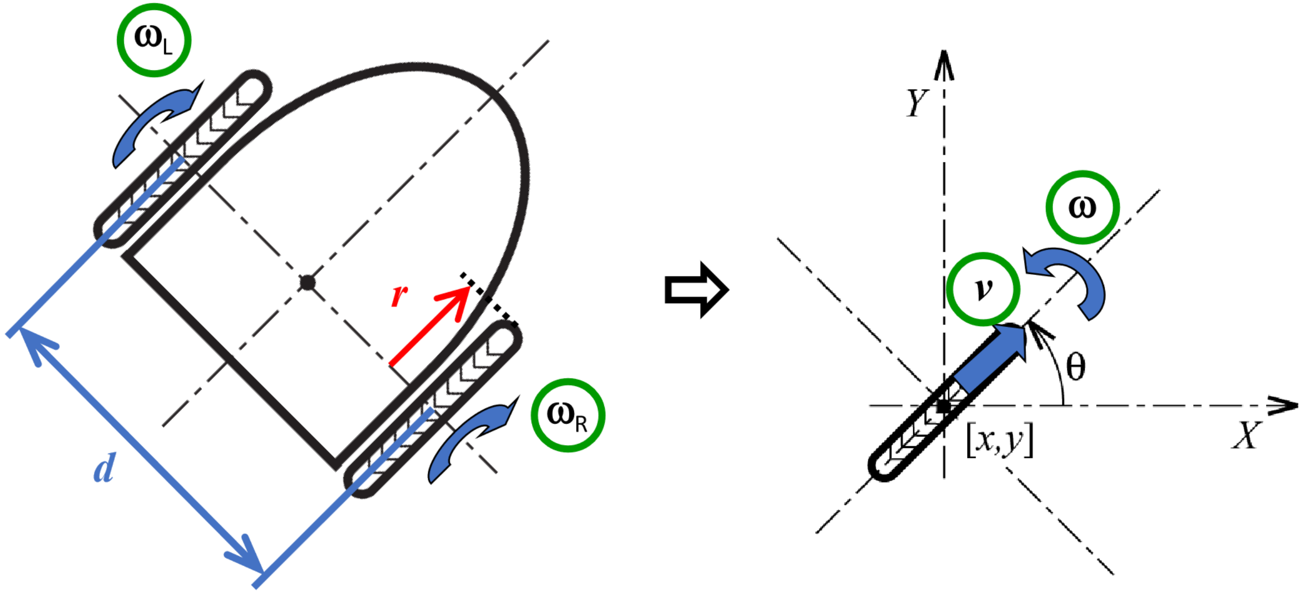
$$v = \omega R$$

$$v = \frac{r(\omega_R + \omega_L)}{2}$$

$$\omega = \frac{r(\omega_R - \omega_L)}{d}$$

$$\omega_R = \frac{2v + \omega d}{2r}$$

$$\omega_L = \frac{2v - \omega d}{2r}$$



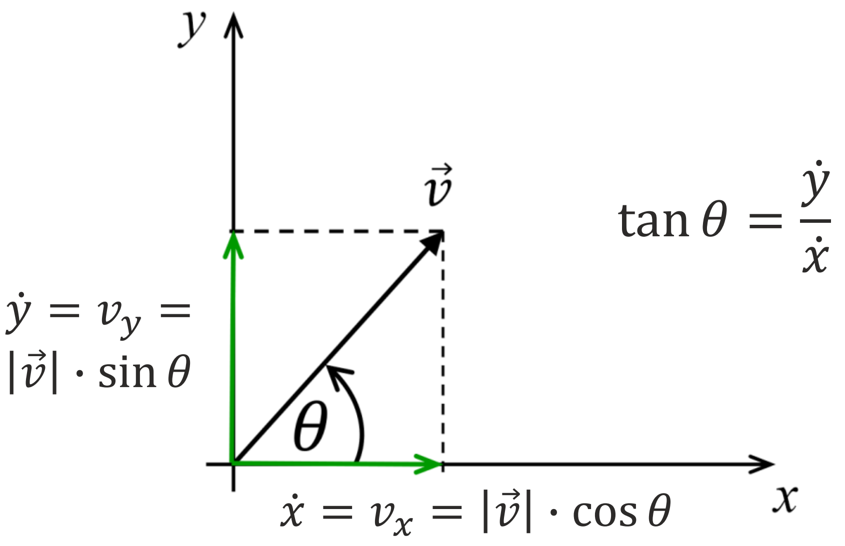
[1]

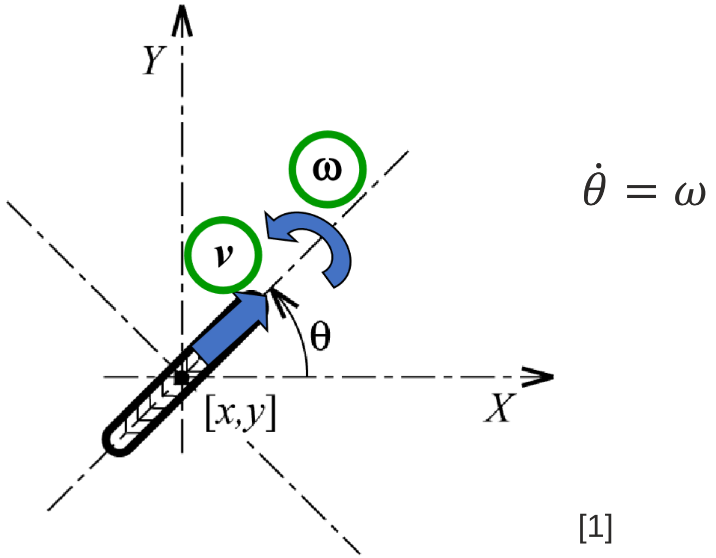
1. HRABEC, Jakub. Kinematika mobilních robotů: Stručný úvod do problematiky. Prezentace pro předmět MPC-RBT. Vysoké učení technické v Brně, 2021.

- State equations
  - 3 states
  - 2 control inputs

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega$$

  - Constraint: $\dot{x}\sin\theta = \dot{y}\cos\theta$



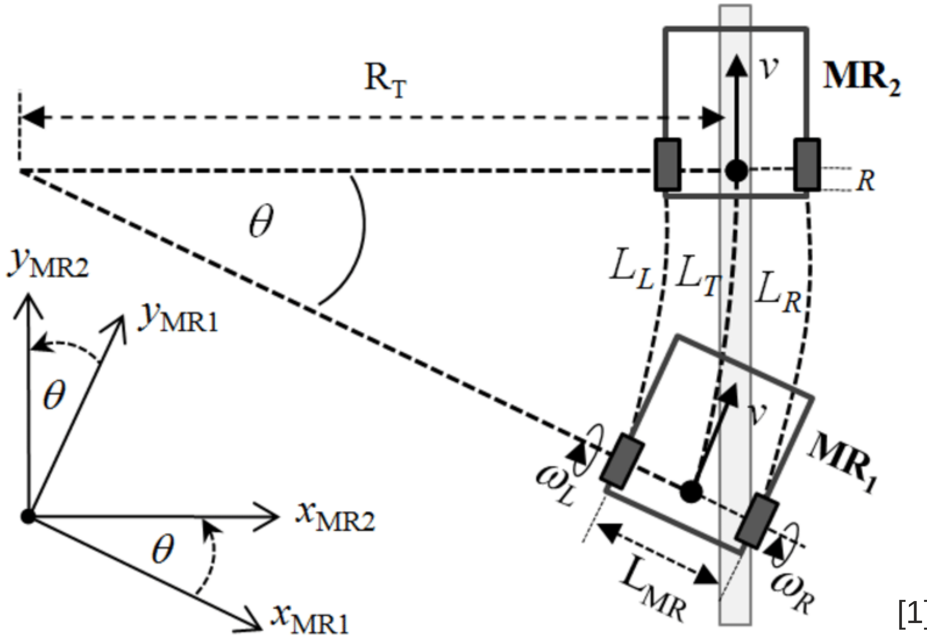


[1]

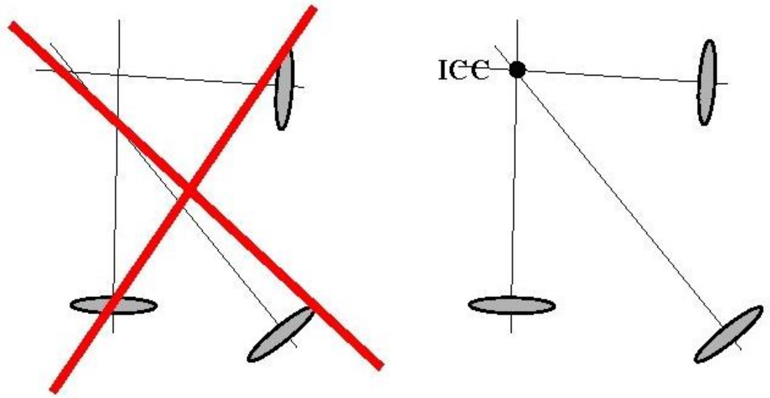
1. HRABEC, Jakub. Kinematika mobilních robotů: Stručný úvod do problematiky. Prezentace pro předmět MPC-RBT. Vysoké učení technické v Brně, 2021.

Differential drive

- Forward kinematics
  - Initial state and control inputs are known
  - What is the final state of the robot?
  - Straightforward task
- Inverse kinematics
  - Initial and final states are known
  - What are the control inputs?
  - Problematic in case of non-holonomic platforms
  - Singular points
- Instantaneous center of curvature (rotation)
  - Each wheel must rotate along its y-axis
  - Wheels move along circular (R < ∞) or straight trajectories



[1]



[2]

1. MAULANA, Eka, M. Aziz MUSLIM a Akhmad ZAINURI. Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot. In: *2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS)* [online]. IEEE, 2014 [cit. 2022-02-14]. ISBN 978-1-4799-6947-0. DOI: 10.1109/EECCIS.2014.7003726
2. BURGARD, Wolfram et al. Probabilistic Robotics: Wheeled Locomotion. Uni Freiburg, 2005.

Differential drive – forward kinematics

- Turning radius

$$R = \frac{v}{\omega} = \frac{\frac{r(\omega_R + \omega_L)}{2}}{\frac{r(\omega_R - \omega_L)}{d}} = \frac{d(\omega_R + \omega_L)}{2(\omega_R - \omega_L)}$$

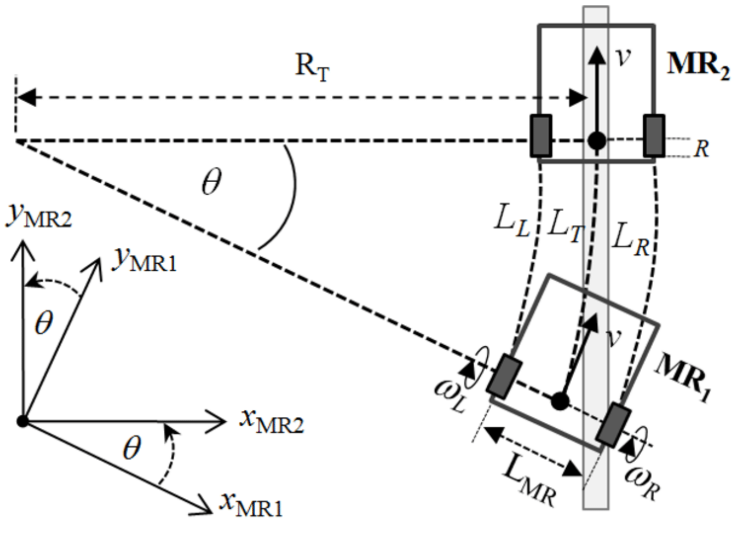
- ICC

$$\begin{pmatrix} ICC_x \\ ICC_y \end{pmatrix} = \begin{pmatrix} x + R\cos(\theta + \pi/2) \\ y + R\sin(\theta + \pi/2) \end{pmatrix} = \begin{pmatrix} x - R\sin\theta \\ y + R\cos\theta \end{pmatrix}$$

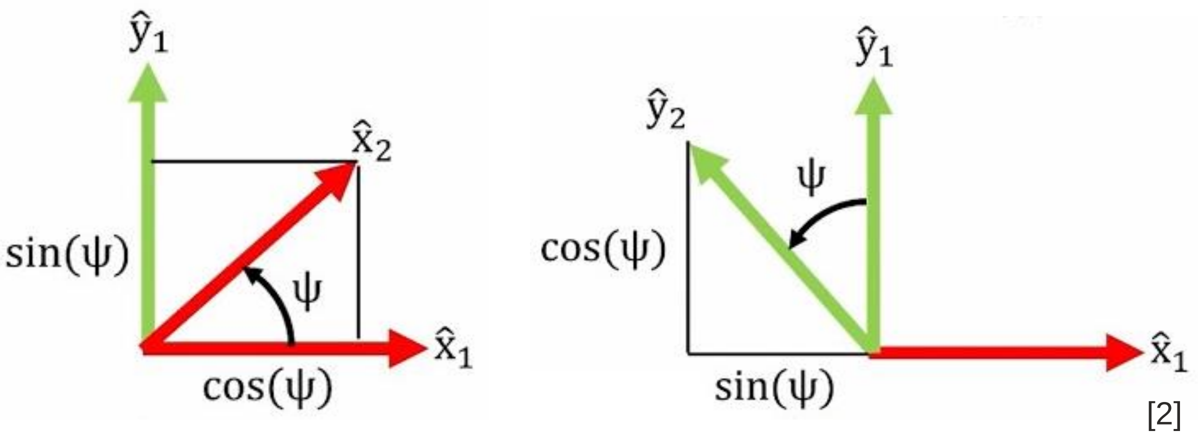
- Rotation matrix

$$\begin{pmatrix} \widehat{x_2} \\ \widehat{y_2} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \widehat{x_1} \\ \widehat{y_1} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \widehat{x_1} \\ \widehat{y_1} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \widehat{x_2} \\ \widehat{y_2} \\ 1 \end{pmatrix}$$



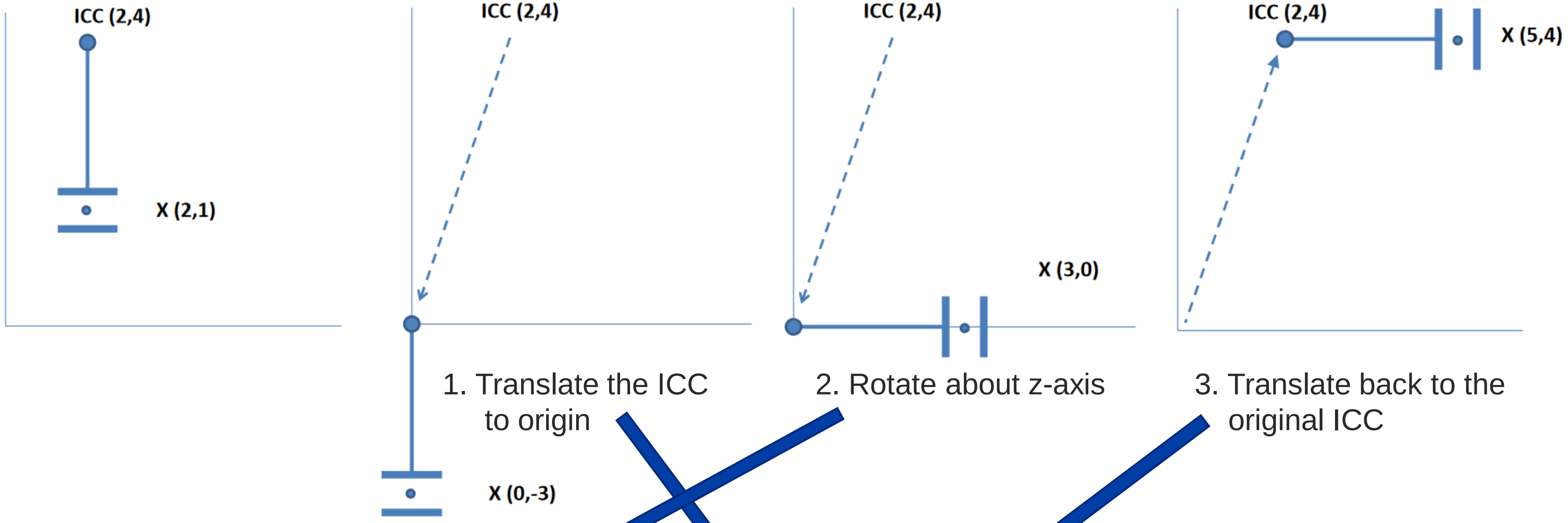
[1]



[2]

1. MAULANA, Eka, M. Aziz MUSLIM a Akhmad ZAINURI. Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot. In: *2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS)* [online]. IEEE, 2014 [cit. 2022-02-14]. ISBN 978-1-4799-6947-0. DOI: 10.1109/EECCIS.2014.7003726
2. WOOLFREY, Jon. 2.2 Rotation Matrices. In: *YouTube* [online]. 2018 [cit. 2022-02-15]. Available at: https://youtu.be/4srS0s1d9Yw

Differential drive – forward kinematics



[1]

1. Translate the ICC to origin
2. Rotate about z-axis
3. Translate back to the original ICC

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{pmatrix} + \begin{pmatrix} ICC_x \\ ICC_y \\ \omega\Delta t \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -R\sin\theta + R\sin(\theta + \omega\Delta t) \\ R\cos\theta - R\cos(\theta + \omega\Delta t) \\ \omega\Delta t \end{pmatrix}$$

1. ALLEN, Peter K. Notes – Differential Drive Robots [online]. Columbia University, 2017 [cit. 2022-02-15]. Available at: http://www.cs.columbia.edu/~allen/F17/NOTES/icckinematics.pdf

Differential drive – inverse kinematics

- We can express the state equations using the rotation matrix

World frame

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = \underbrace{\begin{pmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{pmatrix}}_{=\boldsymbol{R}(\theta)}\begin{pmatrix}\dot{x}_R\\ \dot{y}_R\\ \dot{\theta}_R\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}v\\ 0\\ \omega\end{pmatrix}$$

Robot frame

- Let us invert the equations

$$\begin{pmatrix}v\\ 0\\ \omega\end{pmatrix} = \boldsymbol{R}^{-1}(\theta)\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = \boldsymbol{R}^{\mathsf{T}}(\theta)\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix}$$

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

$$\omega = \dot{\theta}$$

- Ideally, we would set

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = K\begin{pmatrix}x_G - x\\ y_G - y\\ \theta_G - \theta\end{pmatrix}$$

Goal = (x_G, y_G, θ_G)

- But we have to follow the non-holonomic constraint $\dot{x}\sin\theta = \dot{y}\cos\theta$

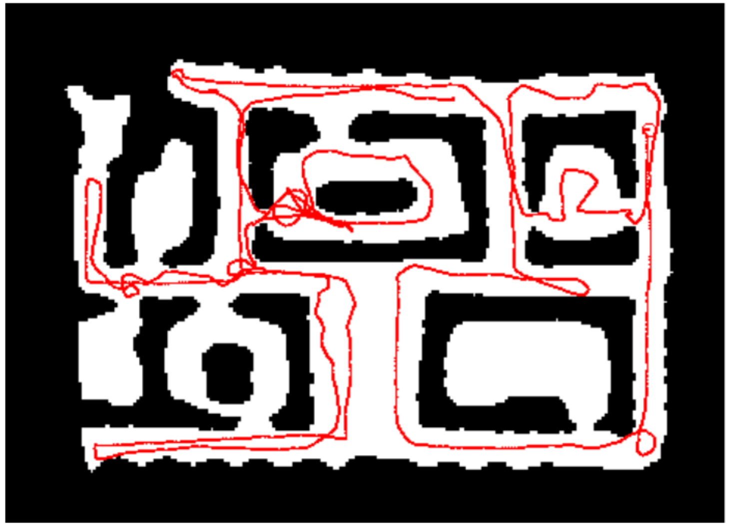
Probabilistic motion model

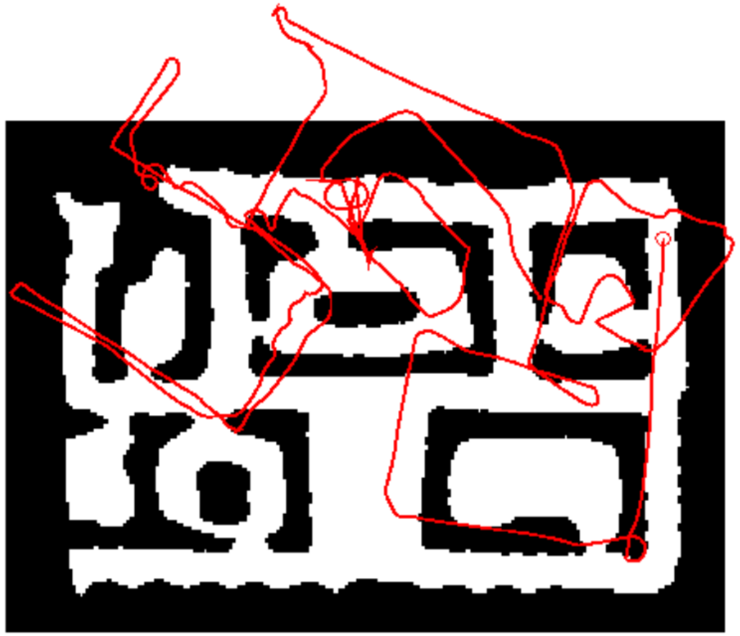
- Motion is inherently uncertain → *probabilistic* models are applied
- Some Bayesian algorithms require a posterior probability $p(x_t|x_{t-1},u_t)$ = what is the probability that action u_t takes the system from state x_{t-1} to state x_t
- Two major types of motion models:
  - Odometry-based – suitable for estimation

$$u_t = \begin{pmatrix} \bar{x}_{t-1} \\ \bar{x}_t \end{pmatrix} \rightarrow (\delta_{\mathrm{rot1}}, \delta_{\mathrm{trans}}, \delta_{\mathrm{rot2}})$$

  - Velicoty-based (dead reckoning) – suitable for prediction

$$u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$$

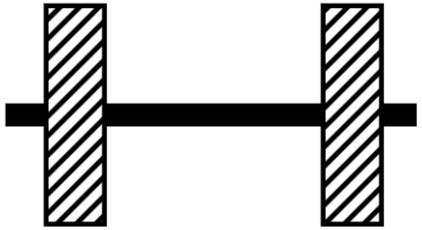




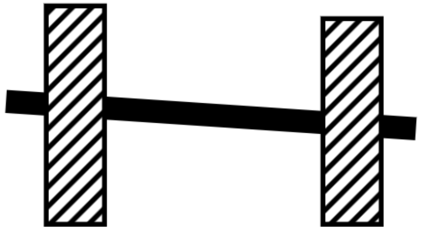
[1]

1. BURGARD, Wolfram et al. Probabilistic Motion Models [online]. Uni Freiburg, 2011 [cit. 2022-02-16]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/slides/06-motion-models.pdf

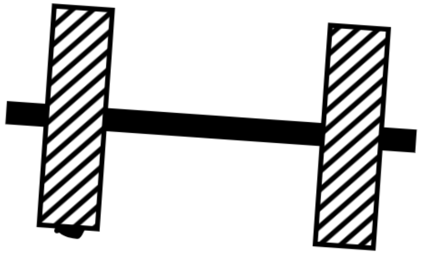
Sources of motion errors



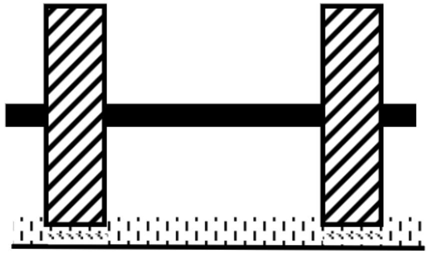
ideal case



different wheel diameters



bump



carpet

[1]

and many more ...

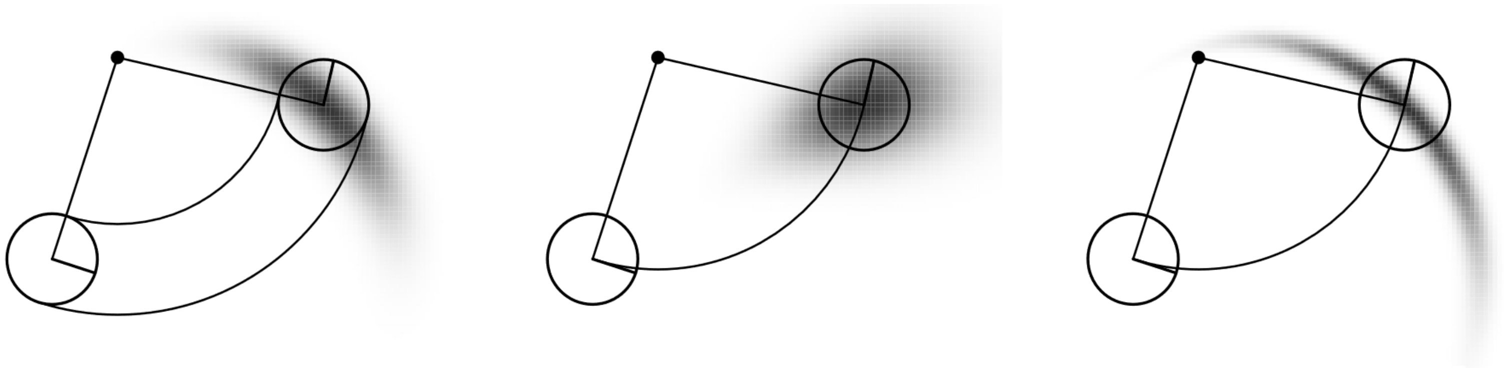
1. BURGARD, Wolfram et al. Probabilistic Motion Models [online]. Uni Freiburg, 2011 [cit. 2022-02-16]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/slides/06-motion-models.pdf

Velocity-based model

- We are looking for $p(x_t|x_{t-1}, u_t)$ for a differential drive
- Naive approach: Normal distribution (3D) centered in x_t = not very realistic
- Let us assume that actual control inputs consist of desired velocities and a noise:

$$\begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix} = \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_1 v^2 + \alpha_2 \omega^2} \\ \varepsilon_{\alpha_3 v^2 + \alpha_4 \omega^2} \end{pmatrix}$$

where ε_{b^2} is a zero-mean error variable with variance b^2,
parameters α_i are robot-specific error coefficients



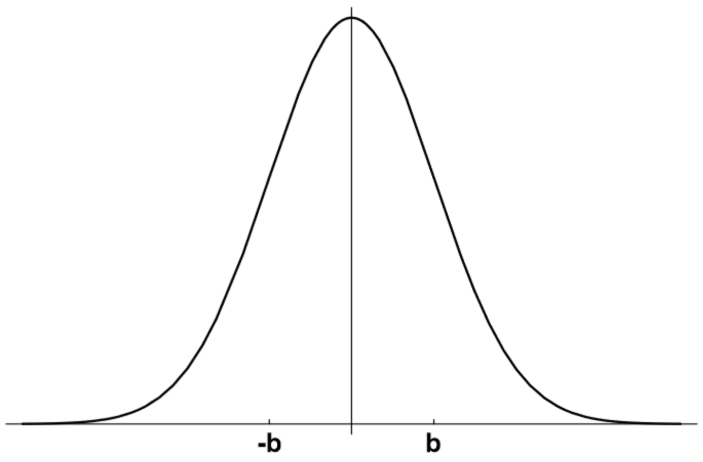
[1]

1. THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. *Probabilistic robotics*. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

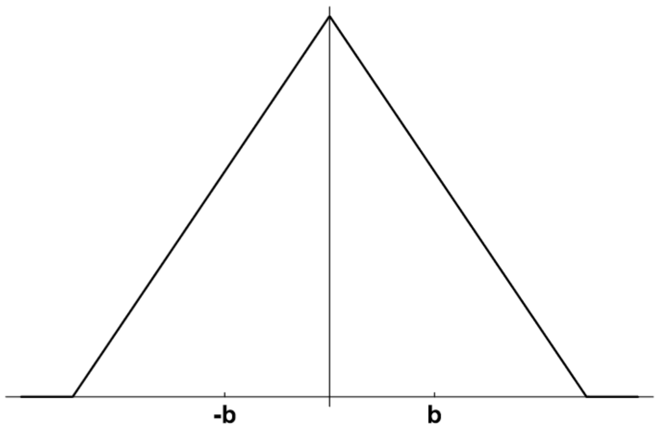
Velocity-based model

- Typical error distributions
  - Normal
  - Triangular
- Problem: 2 velocities control 3 states → all posteriors are located on a 2D manifold in a 3D space → degeneracy
  - Let us add a third variable – ‚final rotation‘ $\hat{\gamma} = \varepsilon_{\alpha_5 v^2 + \alpha_6 \omega^2}$
- Assuming $x_{t-1} = (x, y, \theta)$, $x_t = (x', y', \theta')$, we have

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\hat{v}/_{\hat{\omega}} \sin\theta + \hat{v}/_{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t) \\ \hat{v}/_{\hat{\omega}} \cos\theta - \hat{v}/_{\hat{\omega}} \cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t + \hat{\gamma}\Delta t \end{pmatrix}$$



Normal distribution



[1]

Triangular distribution

1. THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. *Probabilistic robotics*. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

Motion control

Following pre-planned path

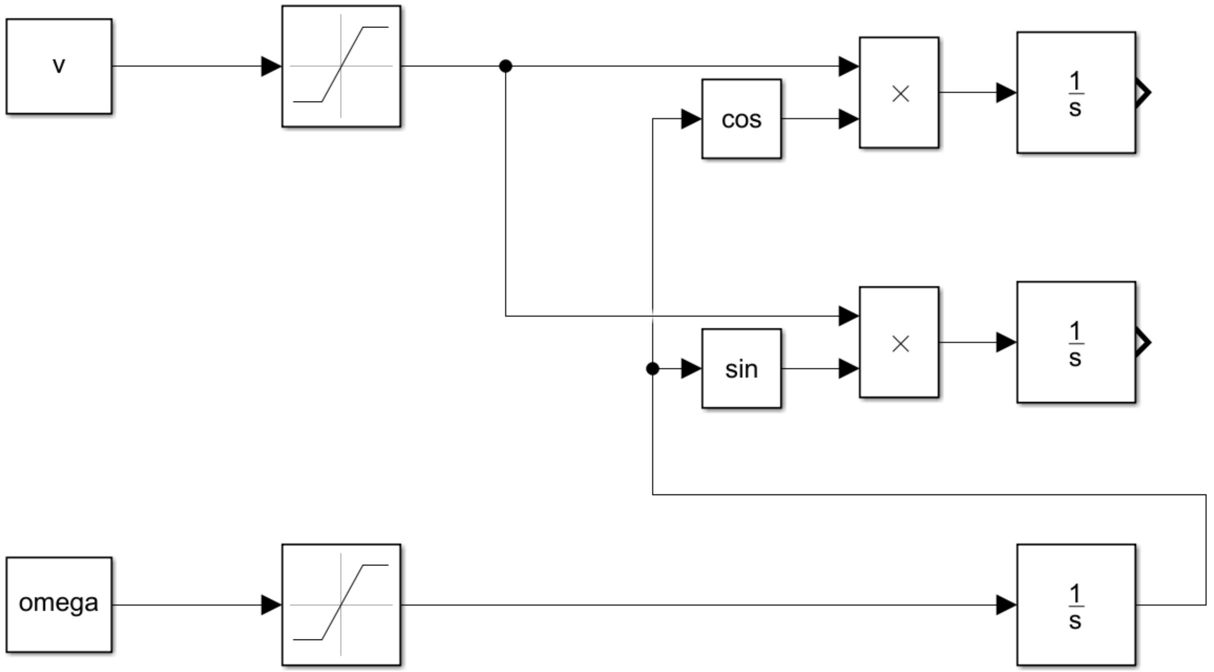
Path following

- Path description
  - Closed-form expression (e.g., Bézier curves)
  - Sequence of waypoints (equidistant vs. adaptive spacing)
- Global planning
  - Optimal path to goal – outlined in *Lecture 6 – Path planning*
- Local planning
  - Optimizing path, considering the motion model
  - Avoiding obstacles – path may be altered to comply with observations
  - Inputs: estimated pose, global path, sensor readings
  - Parameters: kinematic and physical models of the robot, safety requirements
  - Output: local path

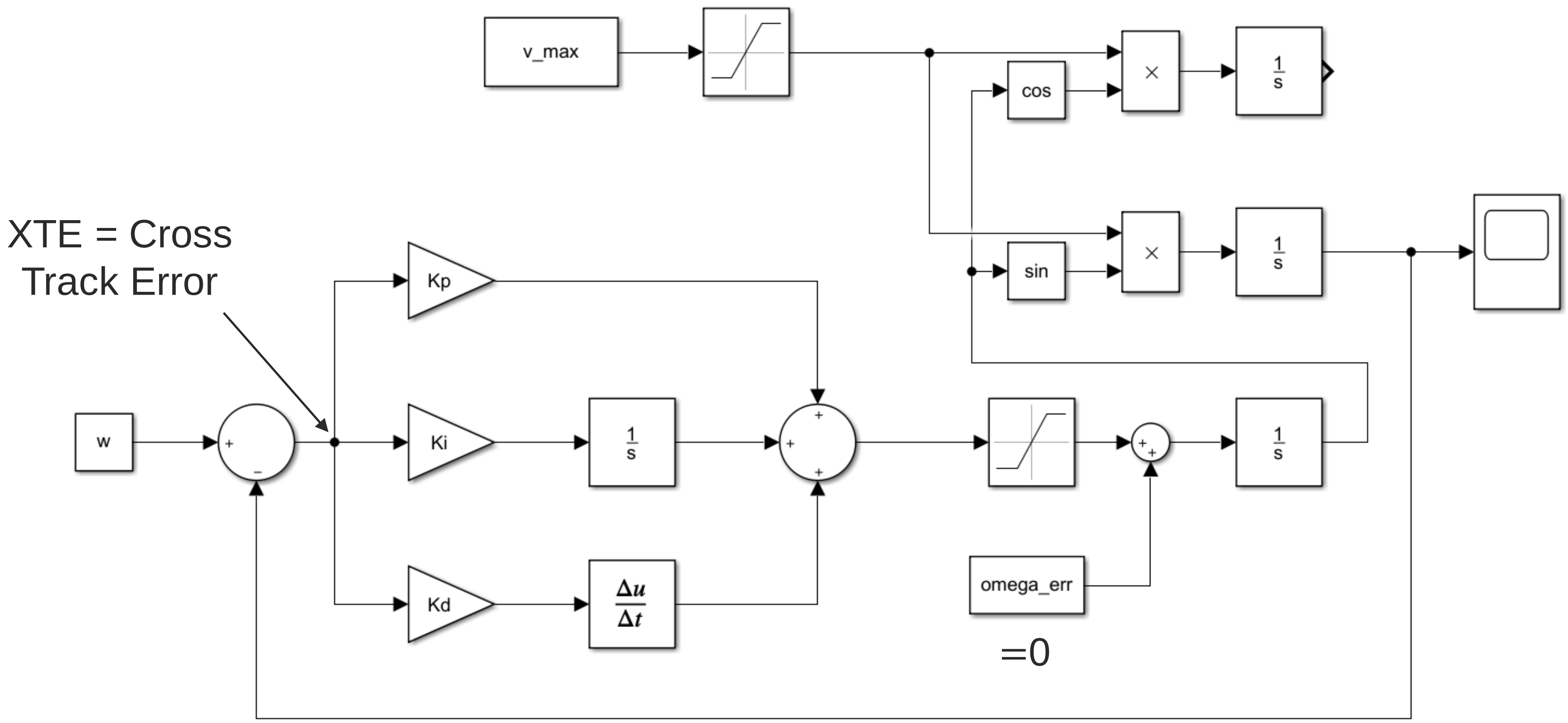
Path following

- **Motion control**
  - Navigate robot along the defined (global/local) path
  - May involve ‚emergency stop‘
  - *Inputs*: estimated pose, desired path, (sensor readings)
  - *Parameters*: kinematic model of the robot
  - *Outputs*: robot control = velocities

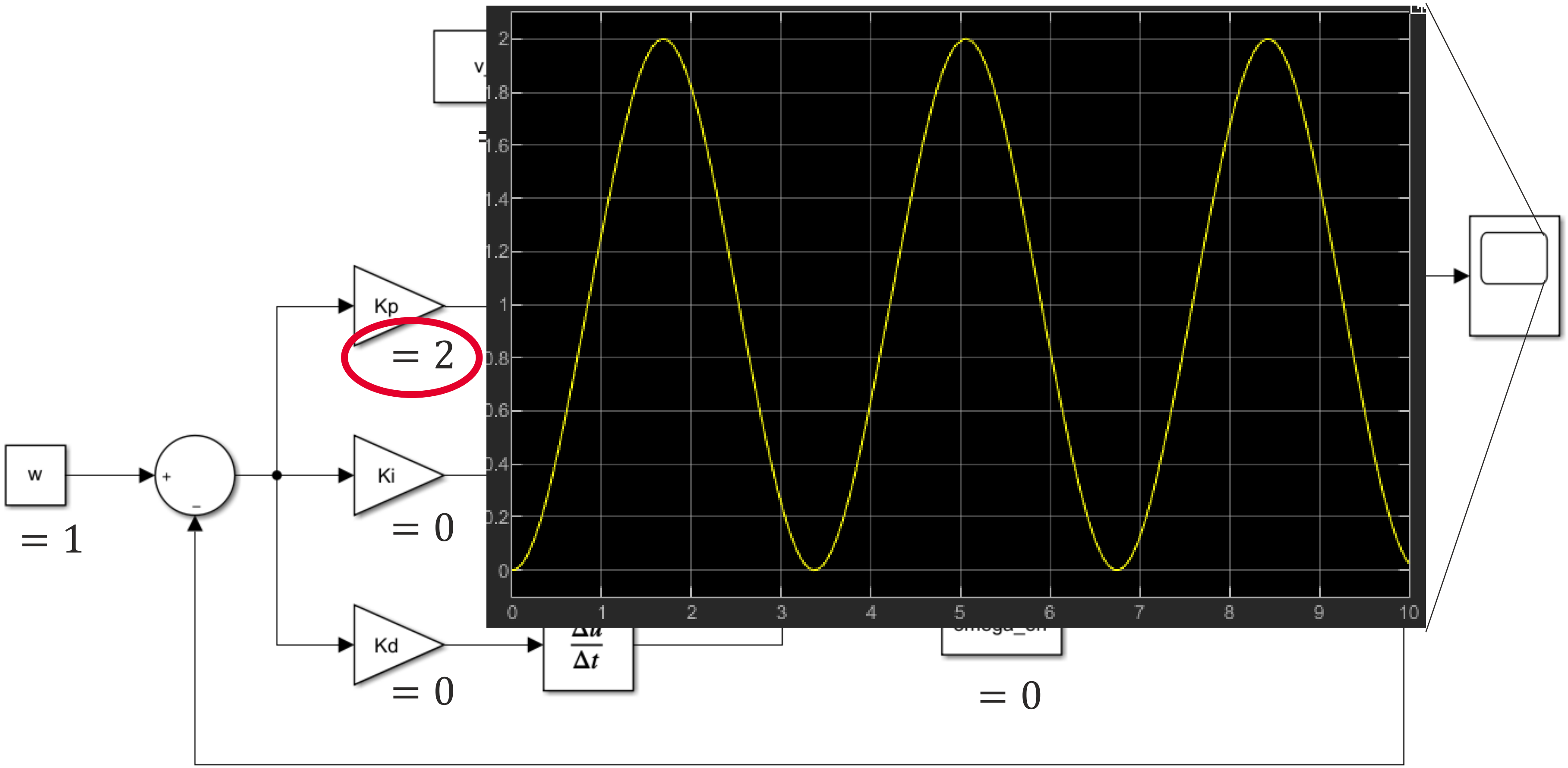
- State model of the differential drive



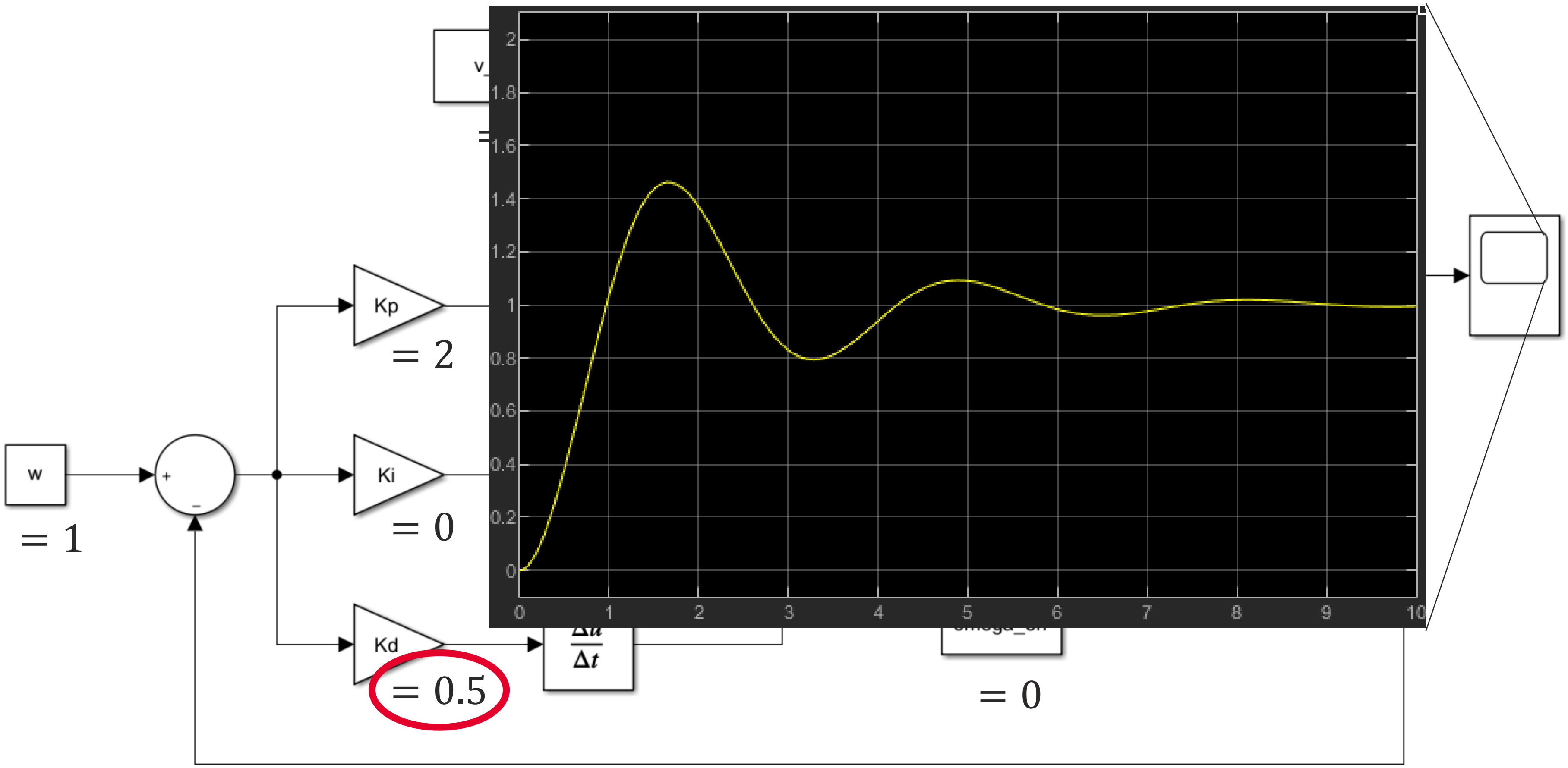
- Let us add a PID structure to control the y state



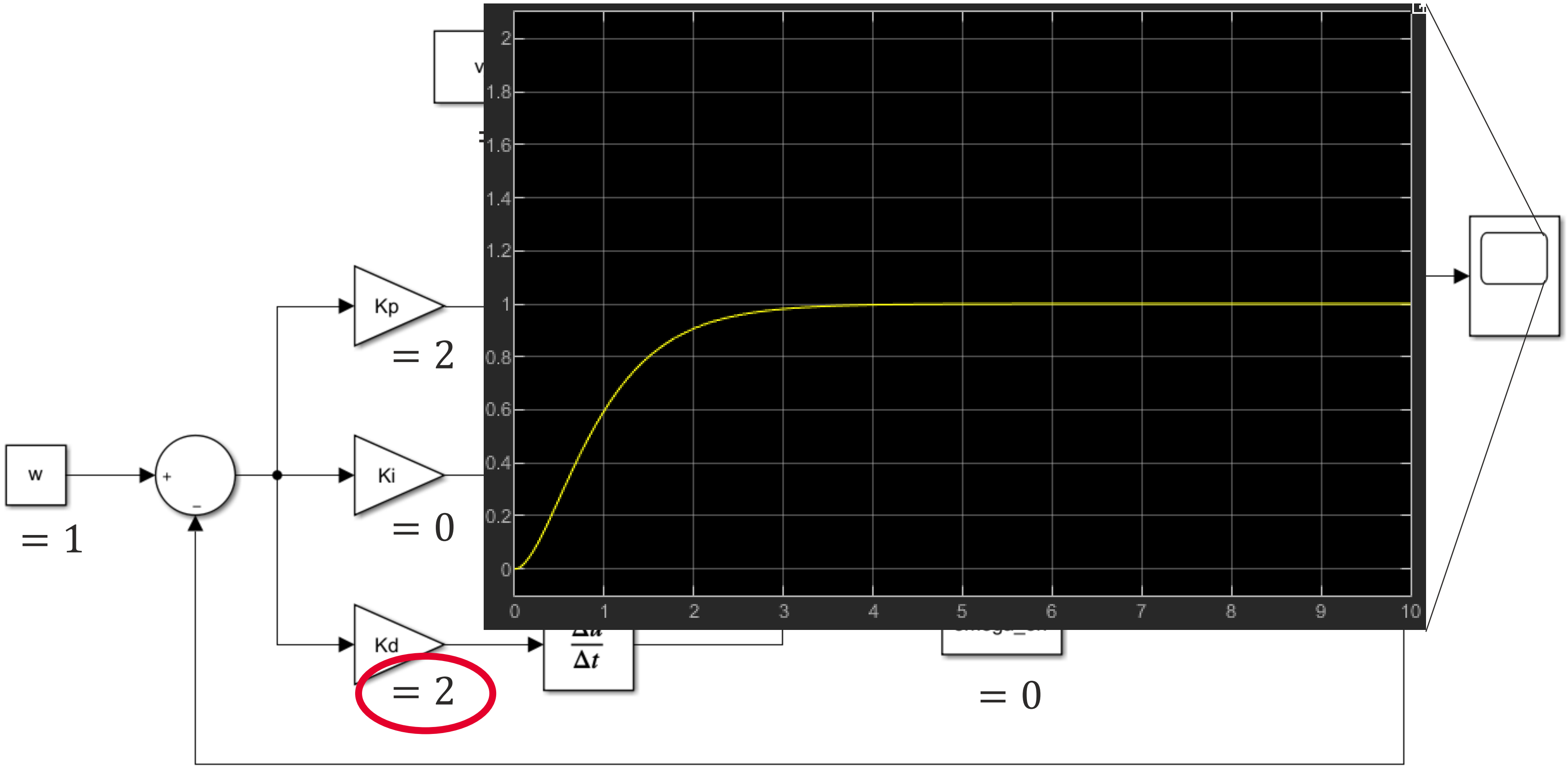
- P controller



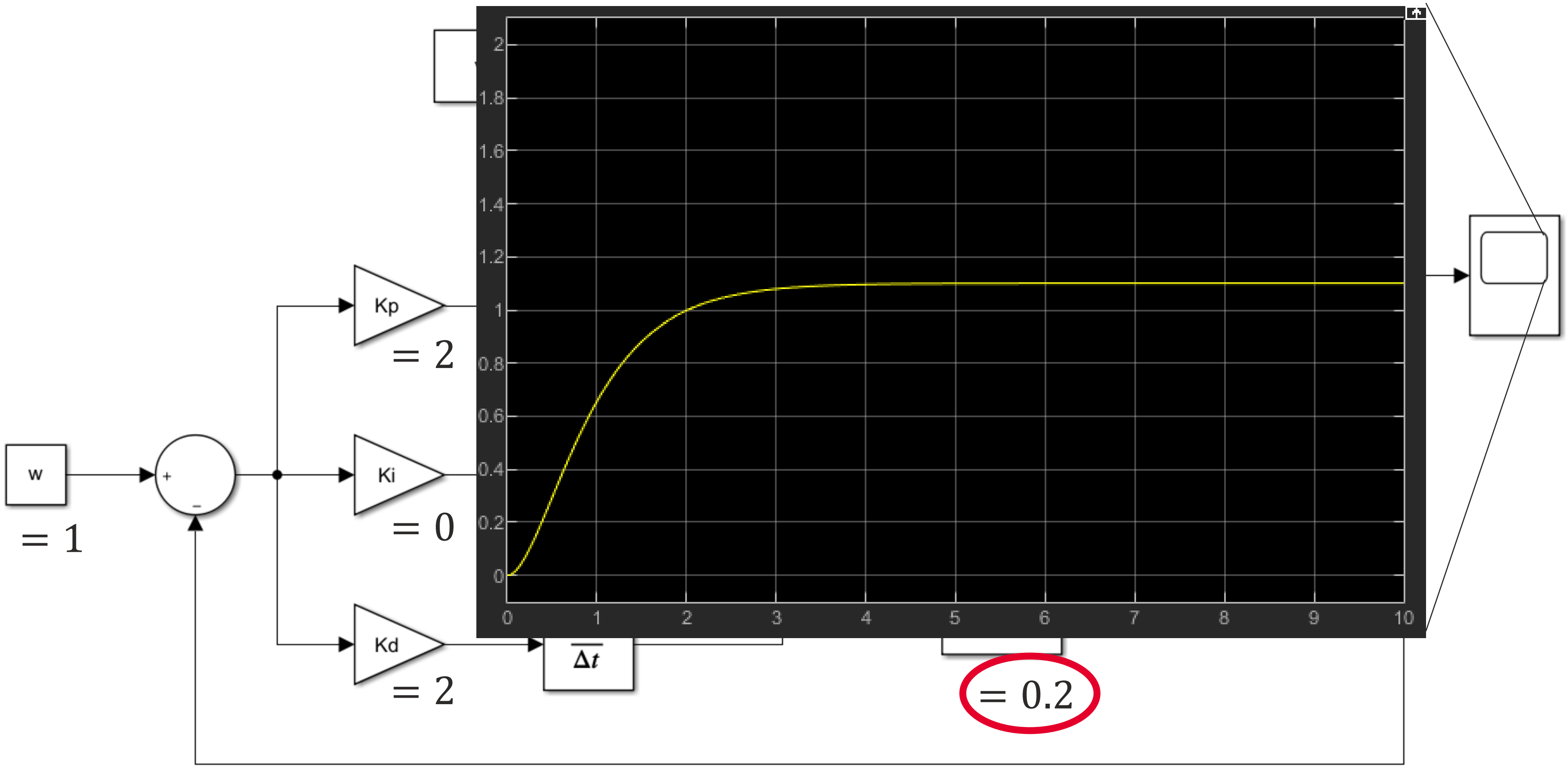
- PD controller



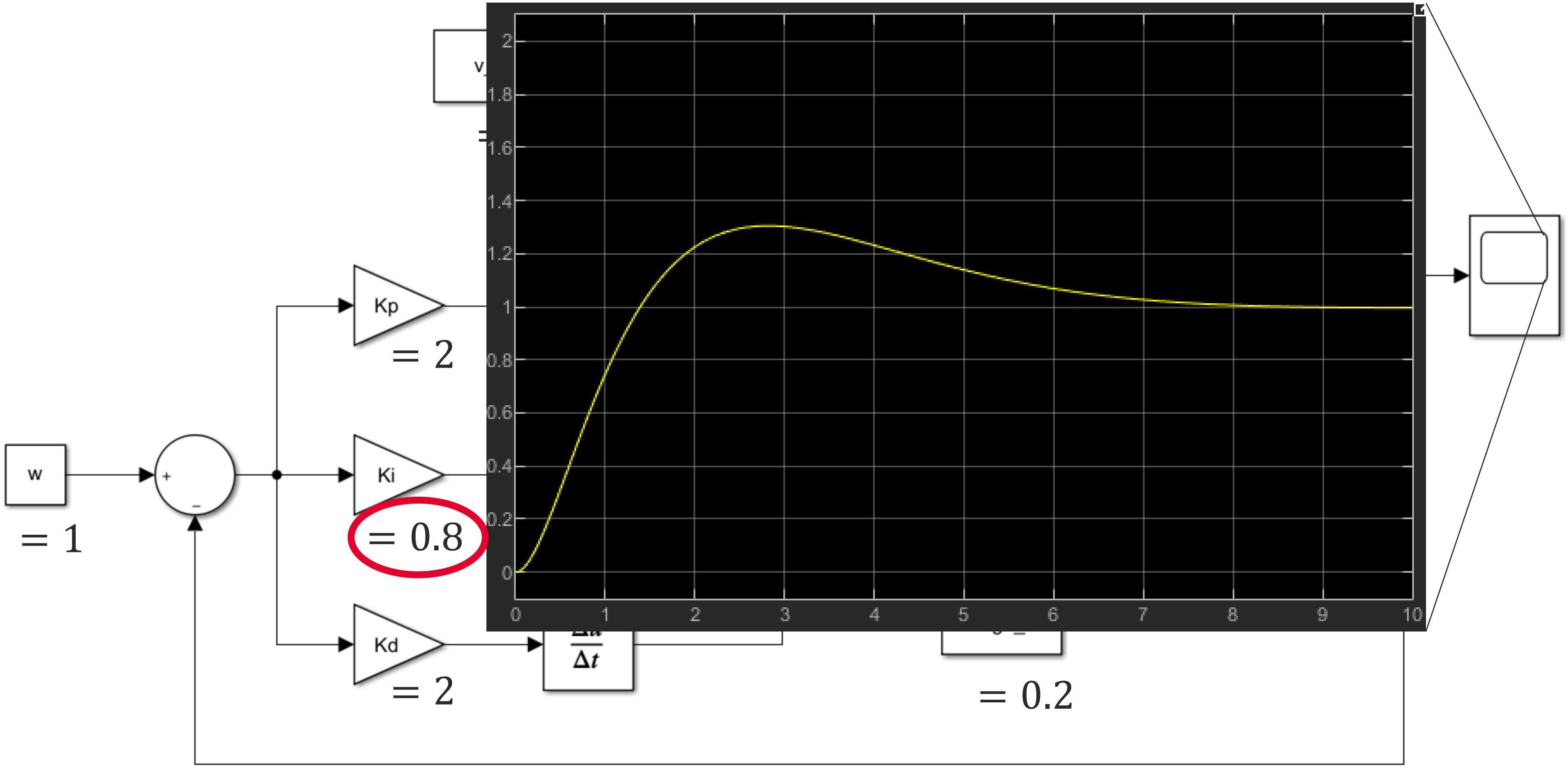
- PD controller – **higher D gain**



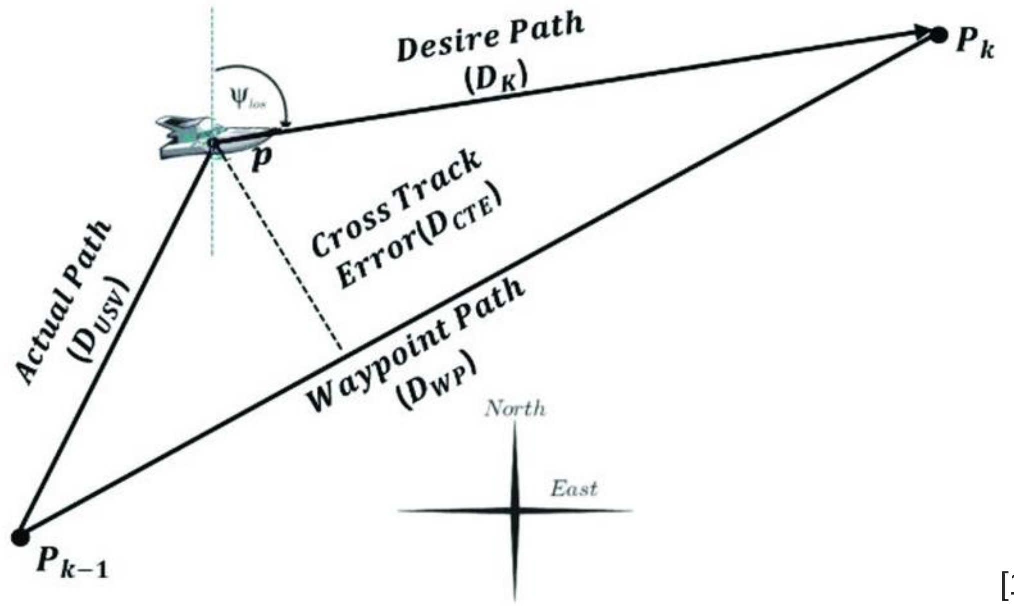
- PD controller – **with added error**



- PID controller – with added error



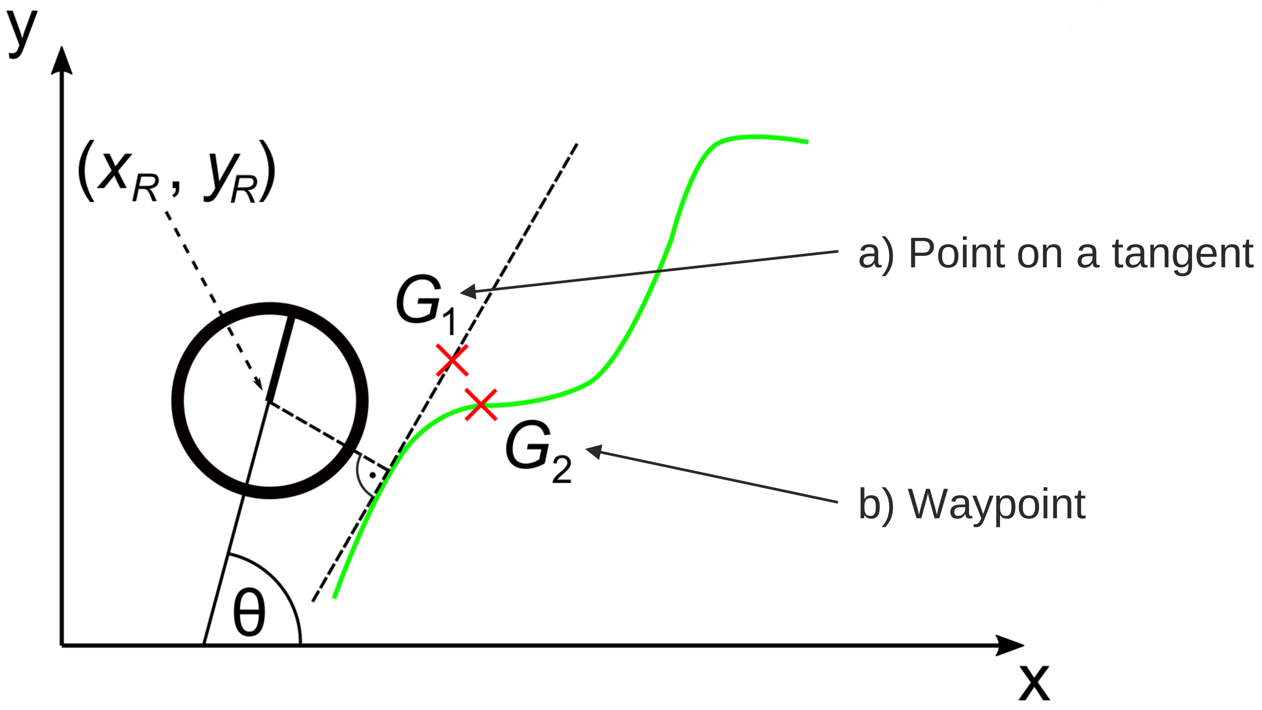
- Possible approaches
  - Cross track error-based
  - Target point-based
  - Others …
- XTE-based
  - Compute cross track error (XTE) = (oriented) shortest distance between the robot and the path
  - Employ arbitrary closed-loop (feedback) controller



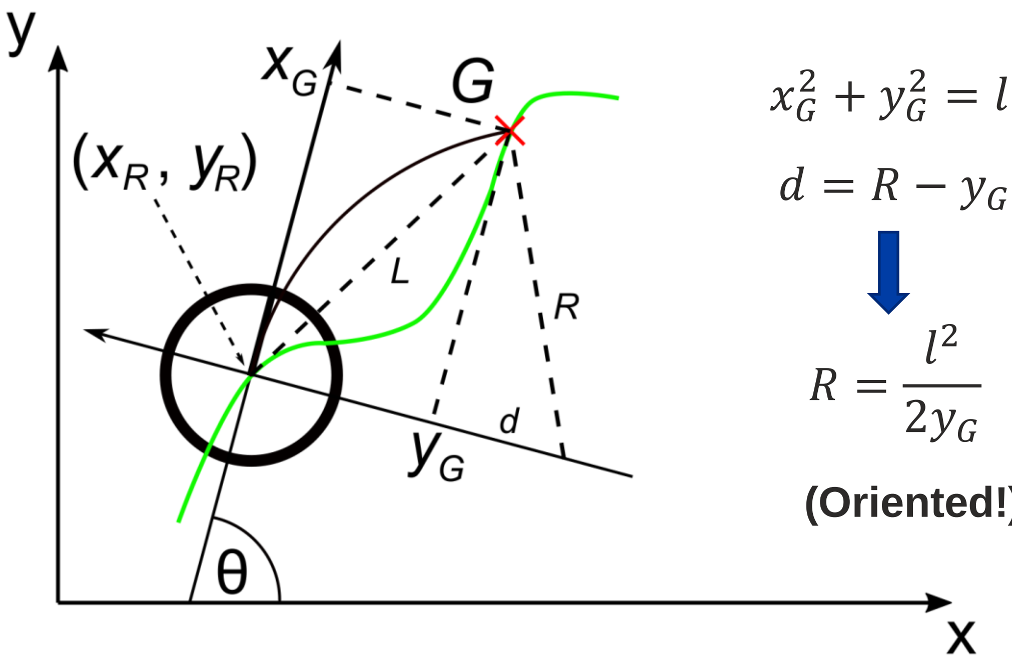
[1]

1. LEE, Ji-Hyeong; JEONG, Sang-Ki; JI, Dae-Hyeong; PARK, Hae-Yong; KIM, Do-Young et al. Unmanned Surface Vehicle Using a Leader–Follower Swarm Control Algorithm. Online. *Applied Sciences*. 2023, vol. 13, issue 5. ISSN 2076-3417. Avaiable at: https://doi.org/10.3390/app13053120. [cit. 2024-02-14].

- Target point-based
  - Identify target (goal) coordinates (x_G, y_G)
    - Waypoint
    - Point on a tangent
  - Pure pursuit path tracking
  - Feedback linearization



- Determine the curvature that will drive the vehicle to the chosen target
- The target is at *lookahead distance* l from the robot
- Constraints: $v_{\max}, \omega_{\max}, R_{\min}$ (minimal turning radius)



$$x_G^2 + y_G^2 = l^2$$

$$d = R - y_G$$

$$R = \frac{l^2}{2y_G}$$

(Oriented!)

a) Ideal case

$$|R| \geq \frac{v_{\max}}{\omega_{\max}} \geq R_{\min}$$

$$\omega = \frac{v_{\max}}{R} < \omega_{\max}$$

$$v = v_{\max}$$

b) Limit forward velocity

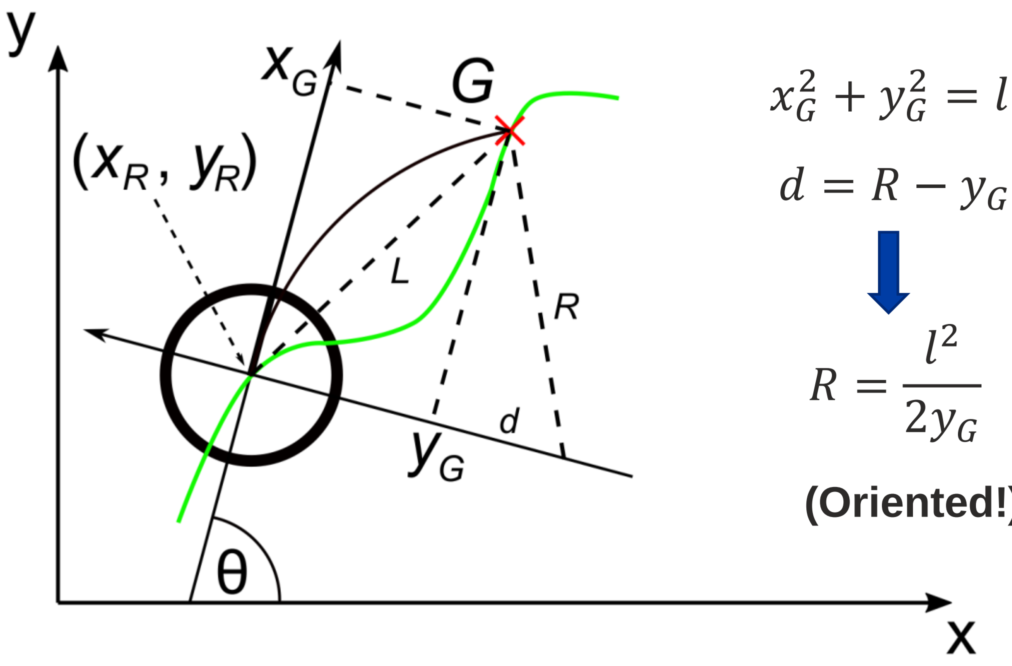
$$\frac{v_{\max}}{\omega_{\max}} > |R| \geq R_{\min}$$

$$\omega = \omega_{\max} \cdot \mathrm{sgn}(R)$$

$$v = |R| \cdot \omega_{\max} < v_{\max}$$

Pure pursuit path tracking

- Determine the curvature that will drive the vehicle to the chosen target
- The target is at *lookahead distance* l from the robot
- Constraints: $v_{\max}, \omega_{\max}, R_{\min}$ (minimal turning radius)



$$x_G^2 + y_G^2 = l^2$$

$$d = R - y_G$$

$$\Downarrow$$

$$R = \frac{l^2}{2y_G}$$

(Oriented!)

c) The goal cannot be reached

$$|R| < R_{\min} \le \frac{v_{\max}}{\omega_{\max}}$$

$$\omega = \omega_{\max} \cdot \mathrm{sgn}(R)$$

$$v = R_{\min} \cdot \omega_{\max} < v_{\max}$$

$$\boxed{R_{\min} \le \frac{v_{\max}}{\omega_{\max}}}$$

- The effect of changing the lookahead distance
- Similar to second order dynamic system (l acts as a damping factor)

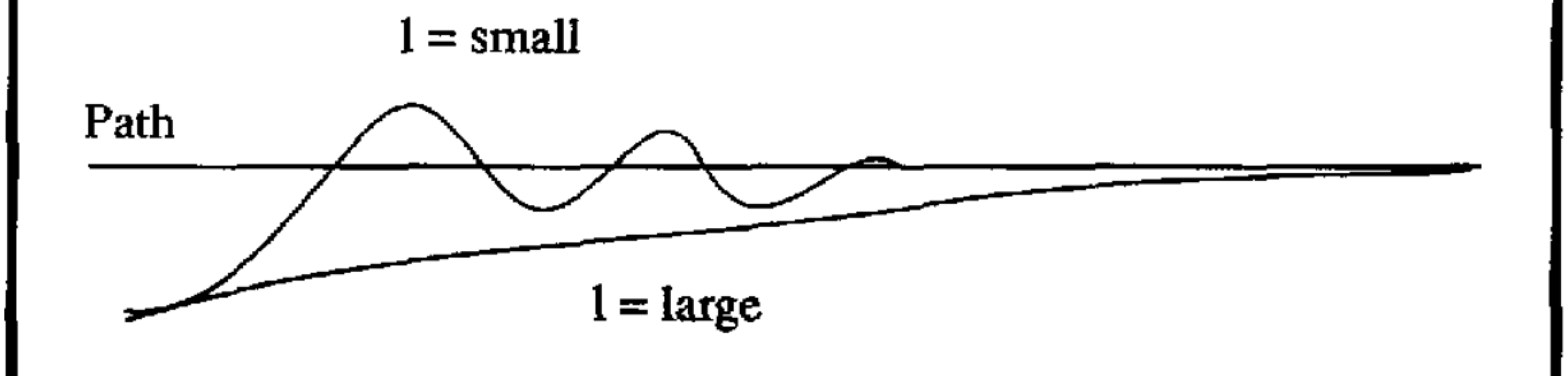


Figure 2. Regaining the path.

[1]

1. COULTER, R. Craig. Implementation of the Pure Pursuit Path Tracking Algorithm [online]. The Robotics Institute, Carnegie Mellon University: 1992 [cit. 2022-02-18]. Available at: https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf

Feedback linearization

- Linear control of holonomic point *P* to control a non-holonomic robot
- Rigid connection of the robot and *P*
- Key idea: $\begin{pmatrix} v \\ \omega \end{pmatrix} = f(\dot{x}_P, \dot{y}_P)$

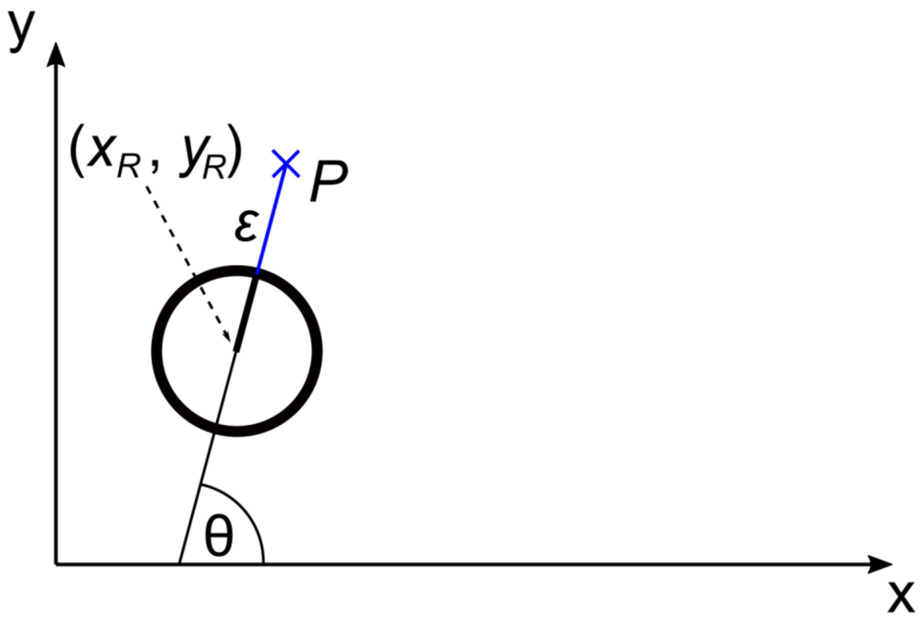
$$x_P = x_R + \varepsilon \cos\theta$$
$$y_P = y_R + \varepsilon \sin\theta$$

$$\xrightarrow{\frac{\mathrm{d}}{\mathrm{d}t}}$$

$$\dot{x}_P = \dot{x}_R + \varepsilon(-\dot{\theta}\sin\theta)$$
$$\dot{y}_P = \dot{y}_R + \varepsilon(\dot{\theta}\cos\theta)$$

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \end{pmatrix} = v\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + \varepsilon\omega\begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}$$

$$v = \dot{x}_P\cos\theta + \dot{y}_P\sin\theta$$
$$\omega = \frac{1}{\varepsilon}(-\dot{x}_P\sin\theta + \dot{y}_P\cos\theta)$$

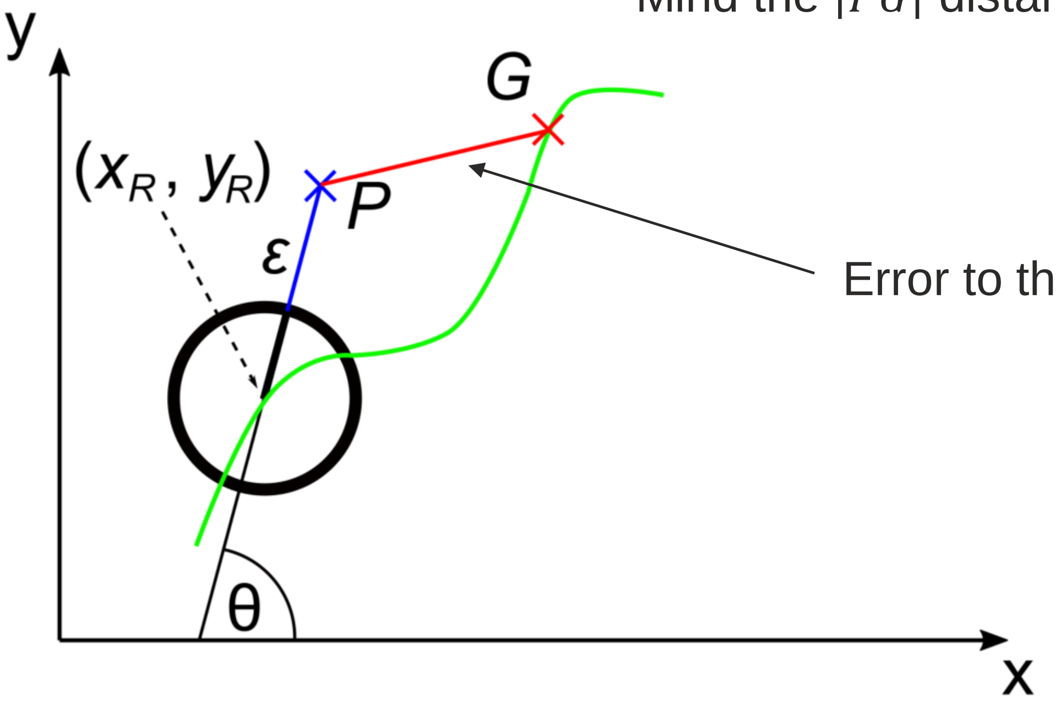


Feedback linearization

If goal is not stationary

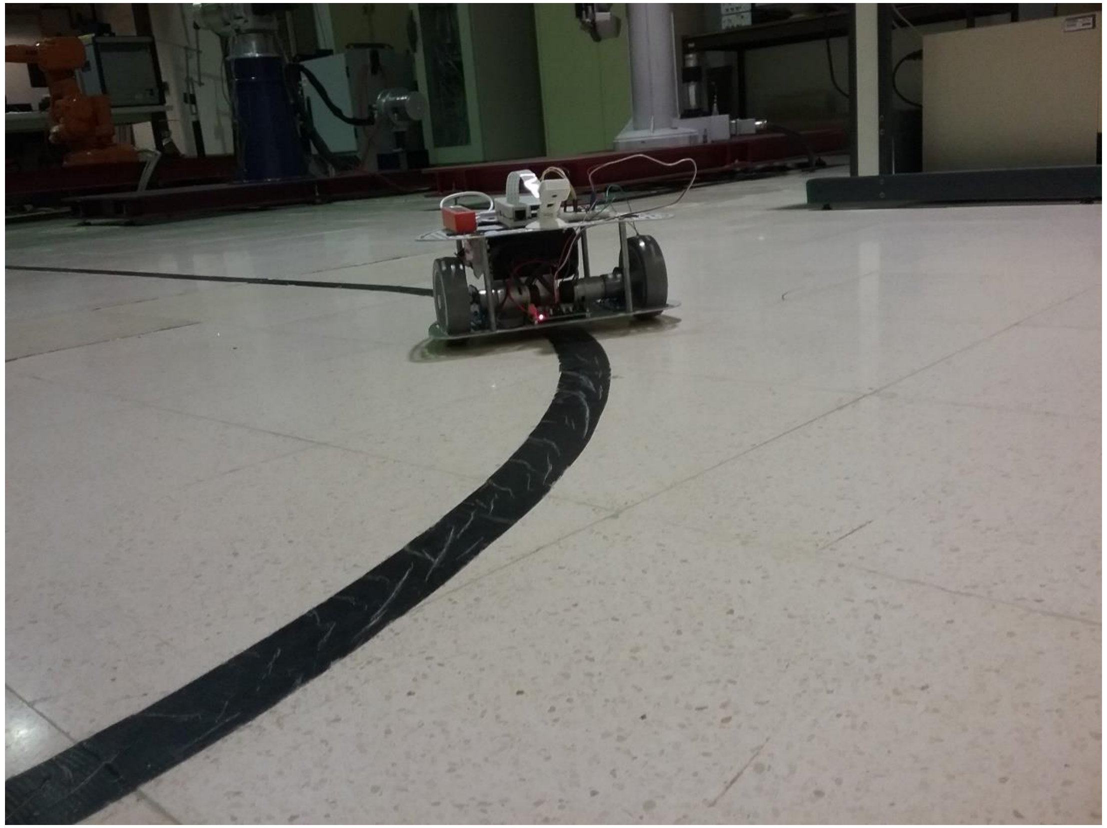
$$\begin{pmatrix}\dot{x}_P\\ \dot{y}_P\end{pmatrix} = \kappa \begin{pmatrix}x_G - x_P\\ y_G - y_P\end{pmatrix} \boxed{+ \begin{pmatrix}\dot{x}_G\\ \dot{y}_G\end{pmatrix}}$$

Mind the $|PG|$ distance



Error to the goal point

- Kinematics
  - Types of drive: differential, Ackermann, omnidirectional, …
  - Holomomic vs. non-holonomic systems/constraints
  - Computing forward and angular velocity from speed of wheels (and vice versa)
  - Forward and inverse kinematics
  - Probabilistic motion models
- Motion control / path following
  - Path description
  - PID controller
  - Pure pursuit tracking algorithm
  - Feedback linearization



[1]

1. SAHA, Anup. Path following robot woth artificial neural network [online]. 2015 [cit. 2022-02-19]. Available at: https://anupmec.wordpress.com/project/hands-on-robot/path-following-robot-with-artificial-neural-network/

Tomas Lazna
tomas.lazna@ceitec.vutbr.cz

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation

Robotics and AI Research Group