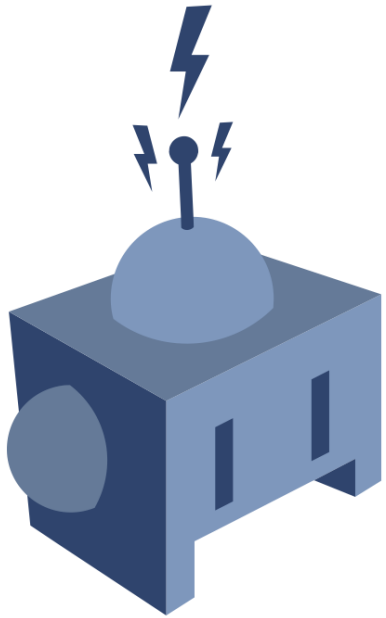
BadgerWorks

Topic: Intro to Modeling & Control for Nonholonomic Systems

Dr. Kostas Alexis (CSE)

Autonomous Robot Challenges

How to model & control Nonholonomic systems



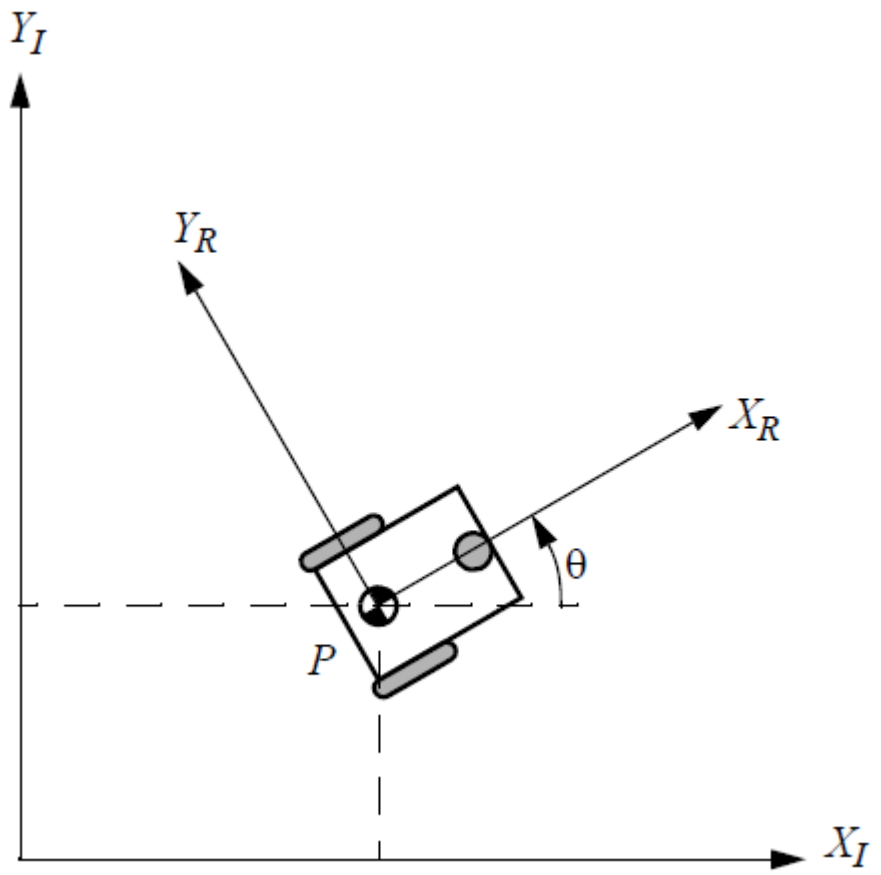
Representing Robot Pose

- The pose of a simplified robot in the 2D configuration space is defined by its x-y coordinates and the heading angle θ.

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- The heading angle of the robot affects its dynamic trajectory in the x-y space. Let us defined the rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



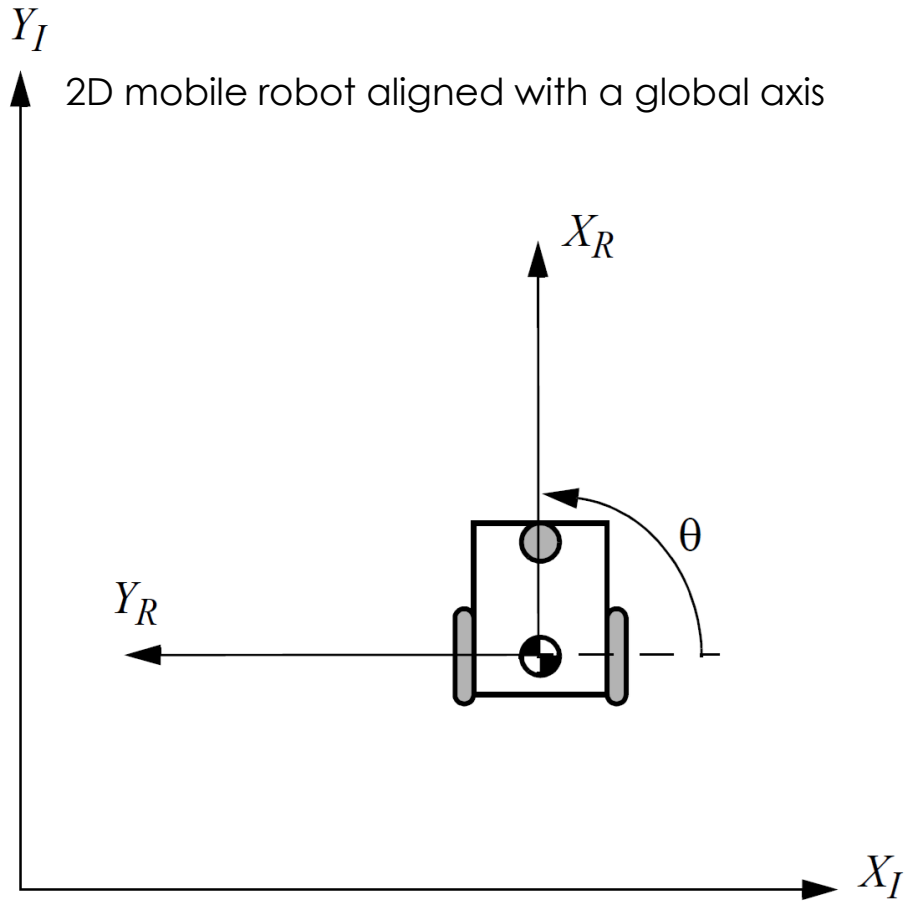
Representing Robot Pose

- The pose of a simplified robot in the 2D configuration space is defined by its x-y coordinates and the heading angle θ.

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- The heading angle of the robot affects its dynamic trajectory in the x-y space. Let us defined the rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Representing Robot Pose

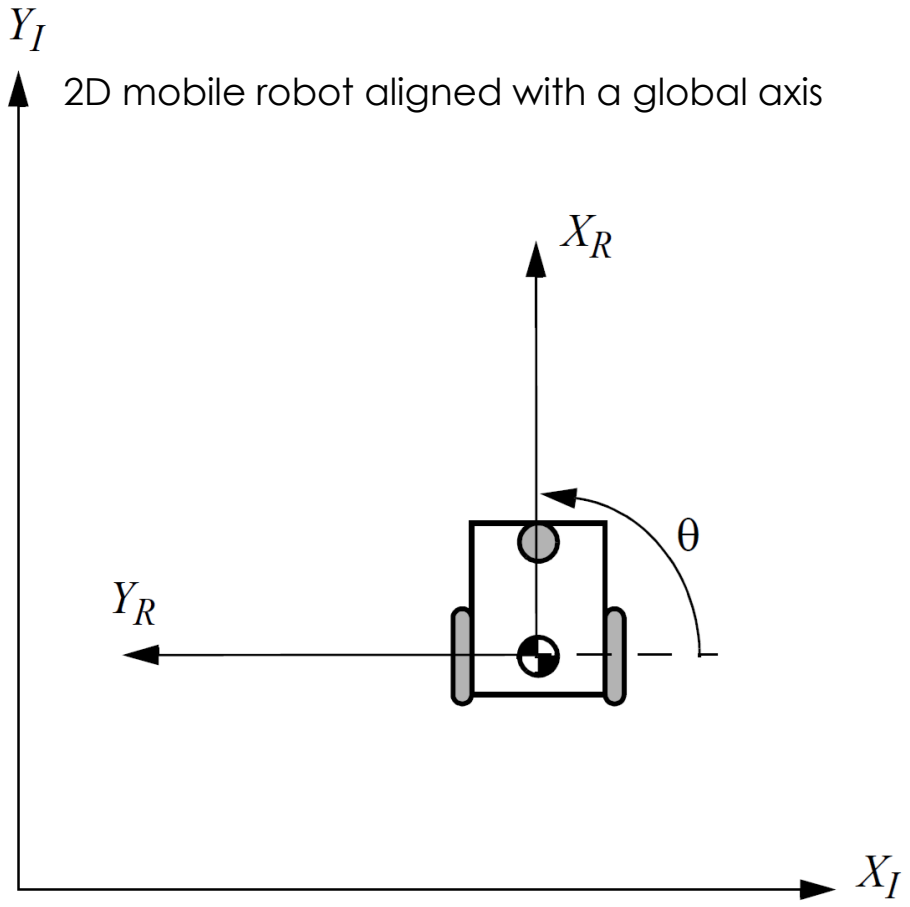
- The rotation matrix $R(\theta)$ can be used to map the motion in the global reference frame (XI, YI) to motion expressed in the local reference frame (XR, YR).

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

- Example for a heading angle of 90 degrees

$$\dot{\xi}_R = R(\frac{\pi}{2})\dot{\xi}_I$$

$$\dot{\xi}_R = R(\frac{\pi}{2})\dot{\xi}_I = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix}$$



Forward Kinematics Model

- In the simplest cases, the equation

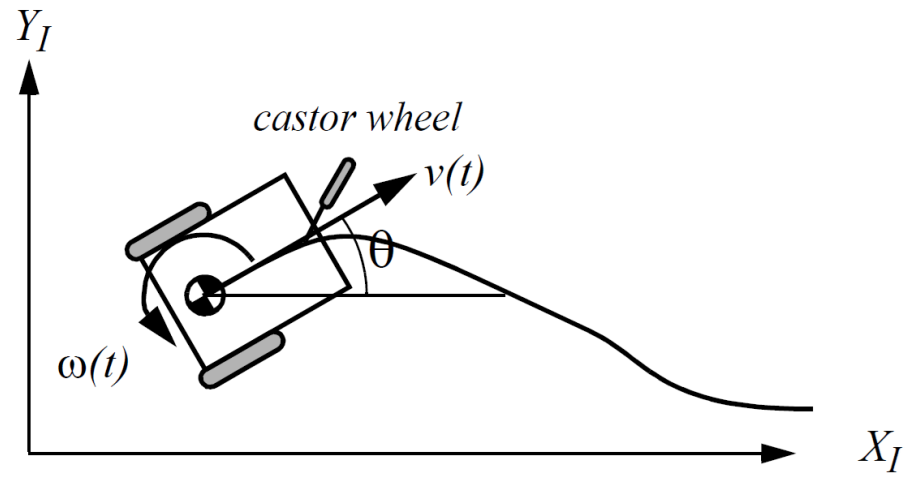
$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

  is sufficient to describe the forward kinematics of a 2D mobile robot

- More generally:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2)$$

- where φ_1, φ_2 the wheel angles (with corresponding turning rates ω_1, ω_2, wheel radius r and distance $2l$ between the two wheels.



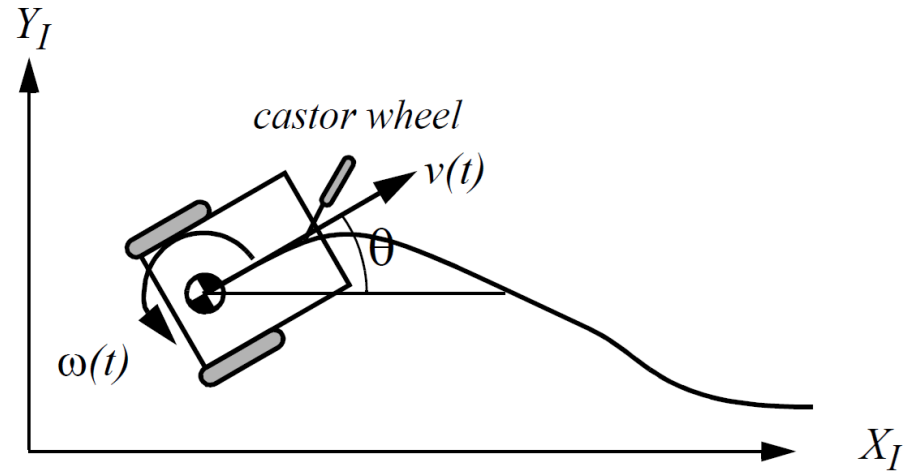
Forward Kinematics Model

- Then:

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \quad \omega_2 = \frac{-r\dot{\varphi}_2}{2l}$$

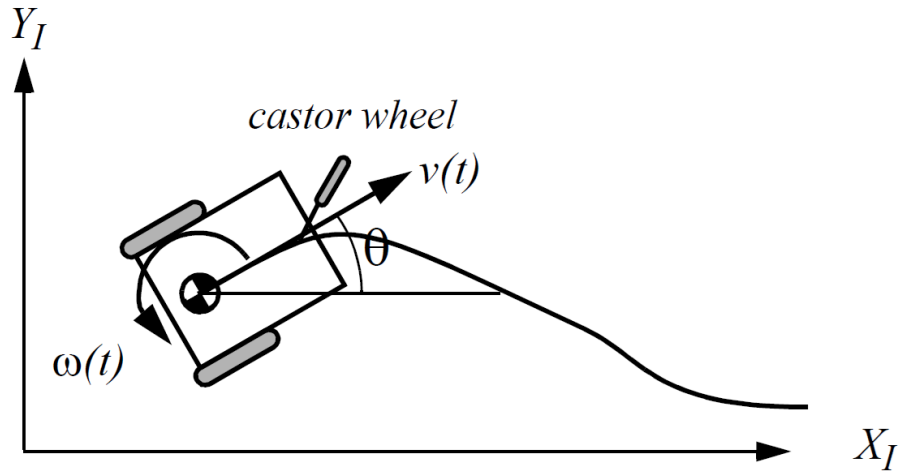
- Therefore:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix}$$



Wheel Kinematic Constraints

- Often wheels have constraints in their motion.
- In our analysis, we still assume certain simplifications, e.g.
  - the plane of the wheel always remains vertical,
  - there is –in all cases- a single point of contact between the wheel and the ground plane,
  - there is no sliding at this single point of contact.

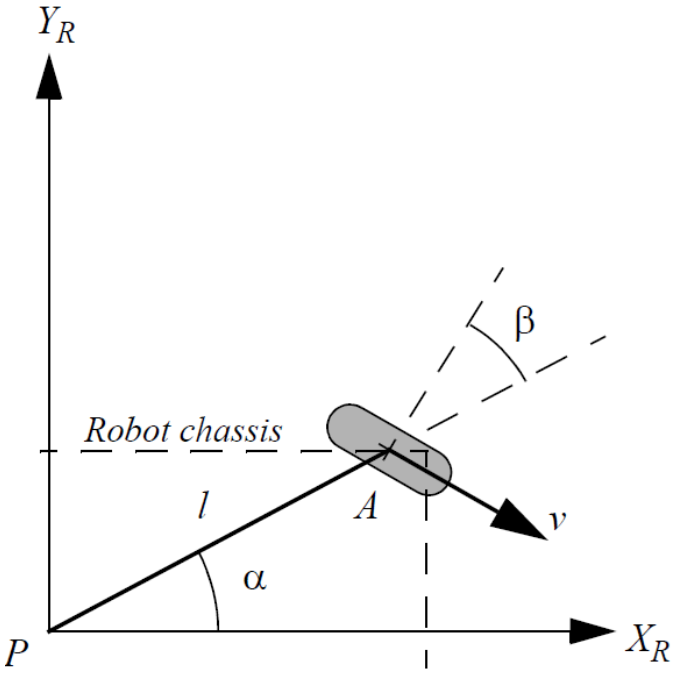


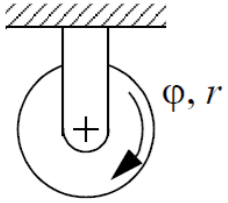
Fixed Standard Wheel

- The **fixed standard wheel** has no vertical axis of rotation for steering.
- Its angle to chassis is fixed and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane.
- Expressing point A at polar coordinates:

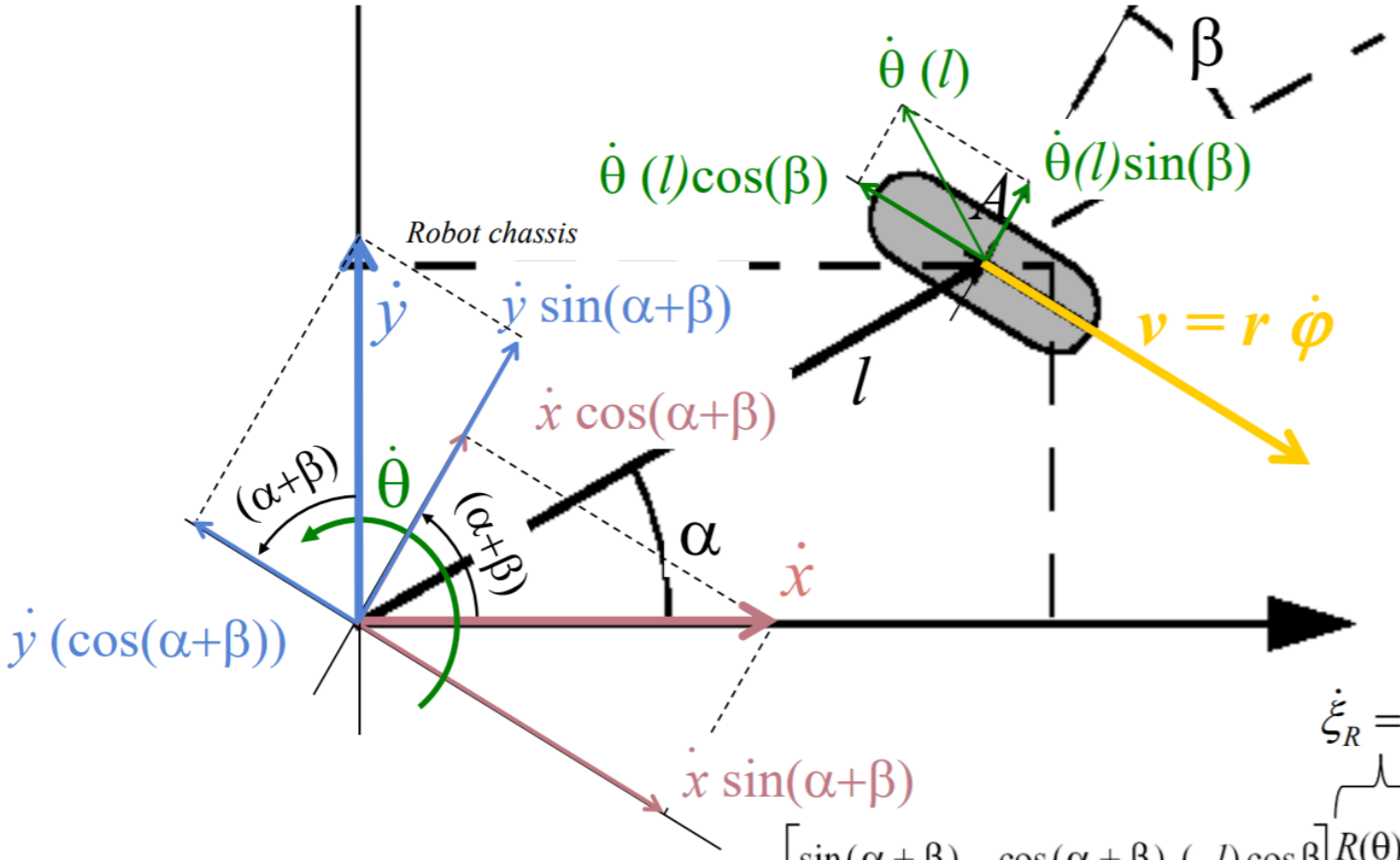
$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

  - The **rolling constraint** for this wheel enforces that all motion along the direction of the wheel plane must be accompanied by the appropriate amount of wheel spin so that there is pure rolling at the contact point.





Fixed Standard Wheel



$$\dot{\xi}_R = \underbrace{\begin{bmatrix}\dot{x} & \dot{y} & \theta\end{bmatrix}^T}$$

$$\begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

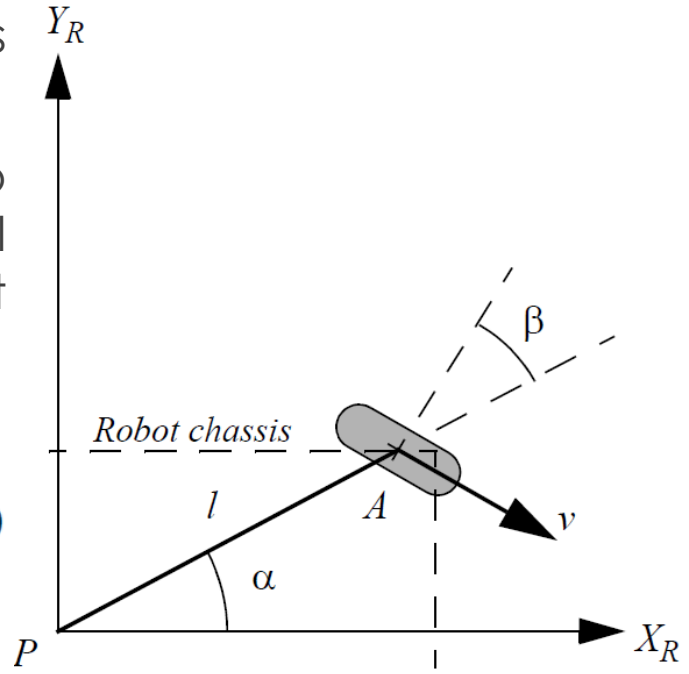
$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix} R(\theta)\dot{\xi}_I = 0$$

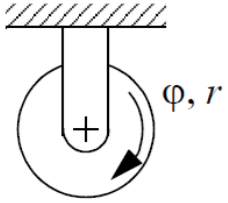
Fixed Standard Wheel

- The fixed standard wheel has no vertical axis of rotation for steering.
- Its angle to chassis is fixed and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane.
- Expressing point A at polar coordinates:

$$\left[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Total motion along the wheel plane





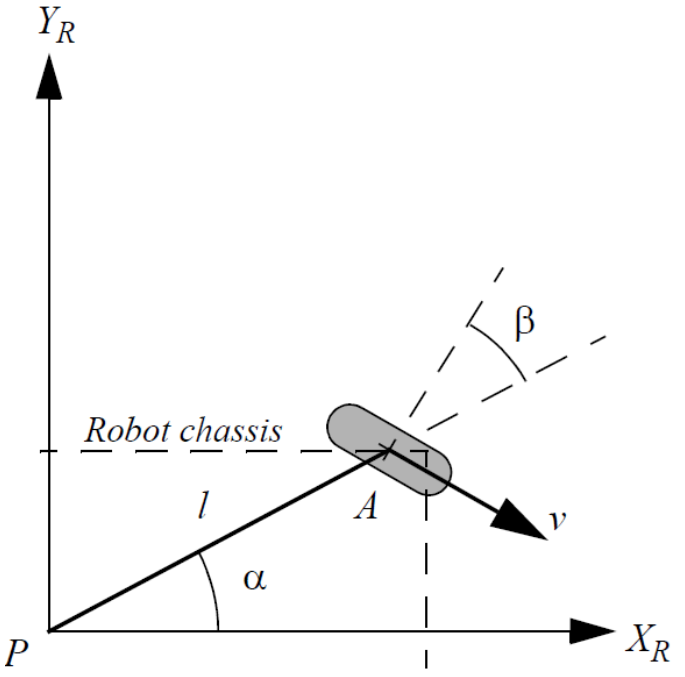
Fixed Standard Wheel

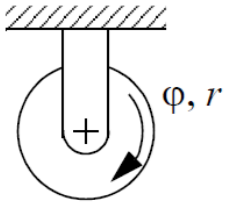
- The fixed standard wheel has no vertical axis of rotation for steering.
- Its angle to chassis is fixed and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane.
- **Expressing point A at polar coordinates:**

$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

- **Sliding constraint** for this wheel enforces that the component of the wheel's motion orthogonal to the wheel plane must be zero:

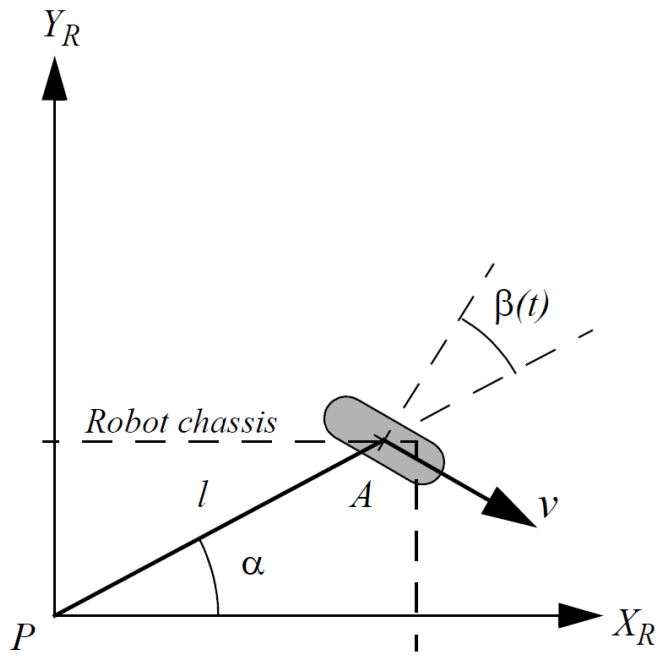
$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; l\sin\beta\right] R(\theta)\dot{\xi}_I = 0$$

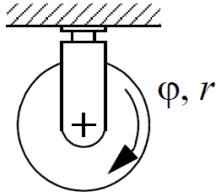




Steered Standard Wheel

- The **steered standard wheel** differs from the fixed standard wheel only in that there is an additional degree of freedom: the wheel may rotate around a vertical axis passing through the center of the wheel and the ground contact point.
- The equations of position for the wheel are similar with the exception that the orientation of the wheel to the robot chassis is no longer a constant β but a function of time $\beta(t)$.





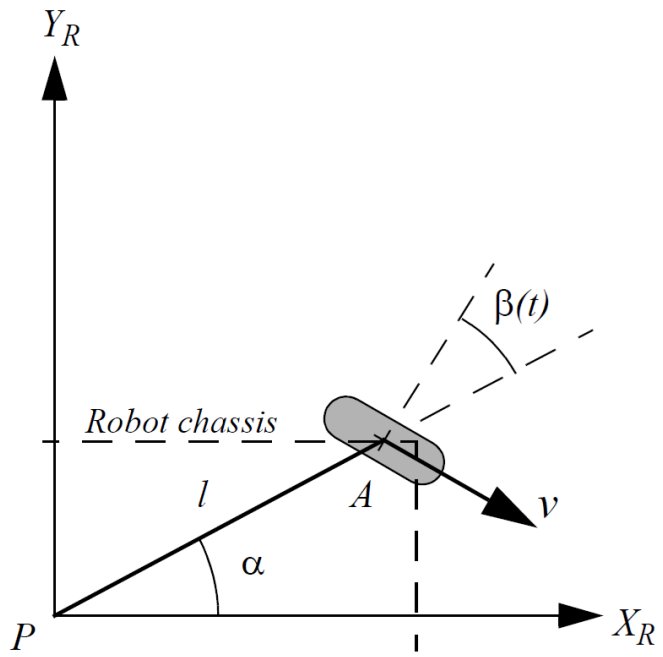
Steered Standard Wheel

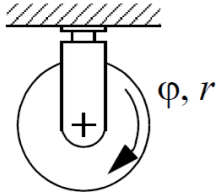
- The **steered standard** wheel differs from the **fixed standard wheel** only in that there is an additional degree of freedom: the wheel may rotate around a vertical axis passing through the center of the wheel and the ground contact point.
- The equations of position for the wheel are similar with the exception that the orientation of the wheel to the robot chassis is no longer a constant β but a function of time $\beta(t)$.
- **Rolling Constraint:**

$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

- **Sliding Constraint:**

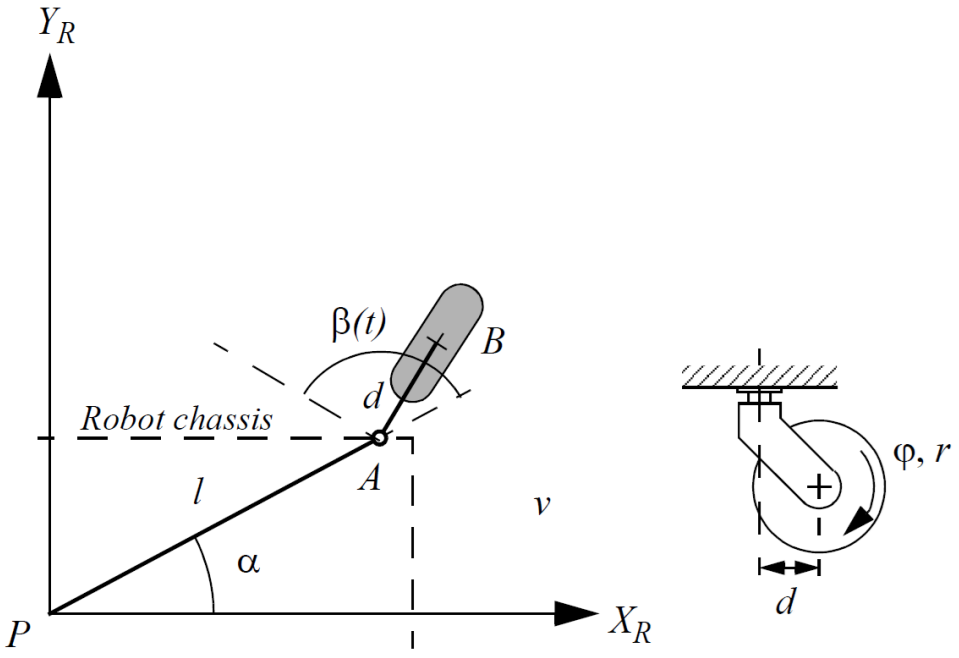
$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; l\sin\beta\right] R(\theta)\dot{\xi}_I = 0\,.$$





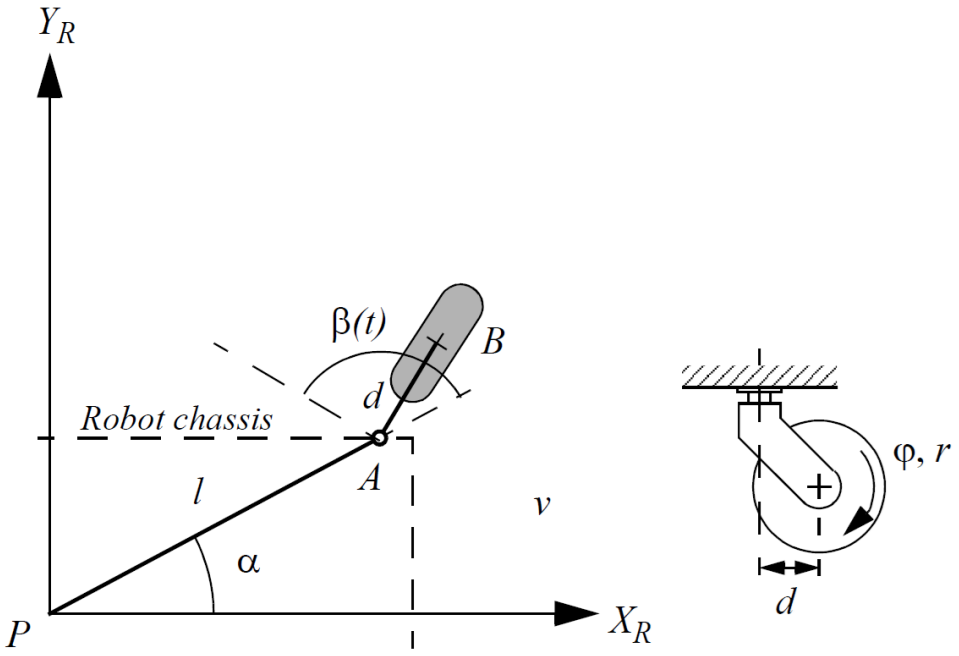
Castor Wheel

- **Castor wheels** are able to steer around a vertical axis but unlike steered standard wheel, the vertical axis of rotation in a castor wheel does not pass through the ground contact point.
- The wheel contact point is not at position B, which is connected by a rigid rod AB of fixed length d to point A, fixes the location of the vertical axis about which B steers, and this point A has a position specified in the robot's reference frame.



Castor Wheel

- **Castor wheels** are able to steer around a vertical axis but unlike steered standard wheel, the vertical axis of rotation in a castor wheel does not pass through the ground contact point.
- The wheel contact point is not at position B, which is connected by a rigid rod AB of fixed length d to point A, fixes the location of the vertical axis about which B steers, and this point A has a position specified in the robot's reference frame.



- **Rolling Constraint:**

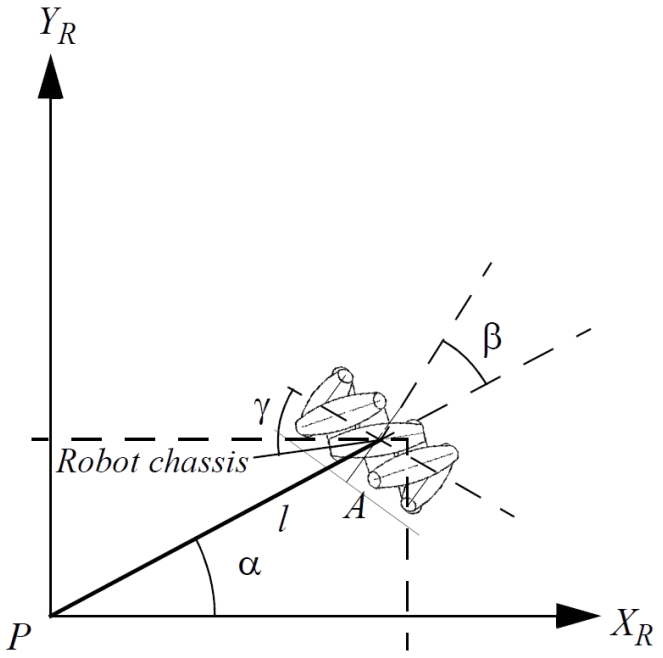
$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

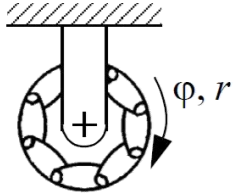
- **Sliding Constraint:**

$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; d + l\sin\beta\right] R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

Swedish Wheel

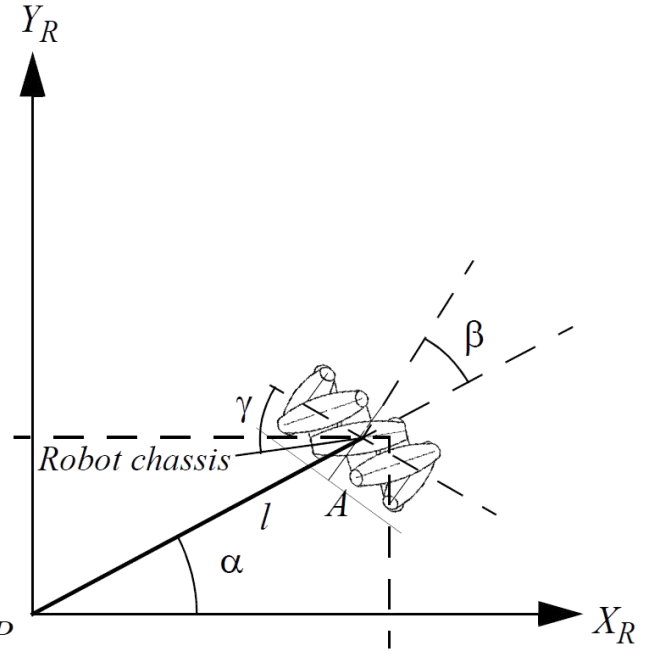
- **Swedish wheels** have no vertical axis of rotation, yet they are able to move omnidirectionally like the castor wheel.
- This is possible by adding a degree of freedom to the fixed standard wheel. Swedish wheels consist of a fixed standard wheel with rollers attached to the wheel perimeter with axes that are antiparallel to the main axis of the fixed wheel component.

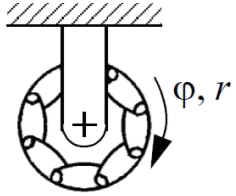




Swedish Wheel

- **Swedish wheels** have no vertical axis of rotation, yet they are able to move omnidirectionally like the castor wheel.
- This is possible by adding a degree of freedom to the fixed standard wheel. Swedish wheels consist of a fixed standard wheel with rollers attached to the wheel perimeter with axes that are antiparallel to the main axis of the fixed wheel component.
- **Motion Constraint:**





$$\left[\sin(\alpha+\beta+\gamma) \;\; -\cos(\alpha+\beta+\gamma) \;\; (-l)\cos(\beta+\gamma)\right] R(\theta)\dot{\xi}_I - r\dot{\varphi}\cos\gamma = 0$$

$$\left[\cos(\alpha+\beta+\gamma) \;\; \sin(\alpha+\beta+\gamma) \;\; l\sin(\beta+\gamma)\right] R(\theta)\dot{\xi}_I - r\dot{\varphi}\sin\gamma - r_{sw}\dot{\varphi}_{sw} = 0$$

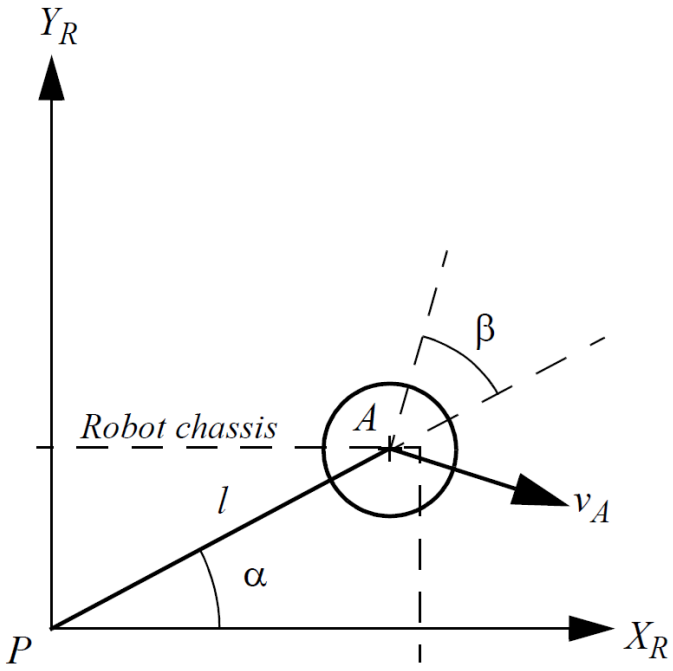
- Angle γ is used such that the effective direction along which the rolling constraint holds is along this zero component and not along the wheel plane

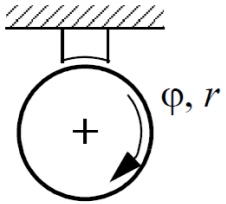
Spherical Wheel

- The **spherical (or ball) wheel** places no constraints on motion. It has no principal axis of rotation.
  - No appropriate rolling and sliding constraints.
- *Motion equation of spherical wheel:*

$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; l\sin\beta\right] R(\theta)\dot{\xi}_I = 0$$





Robot Kinematic Constraints

- Given a robot with **M wheels** we can use the knowledge of how each wheel imposes zero or more constraints on robot motion to form the overall robot kinematic constraints.
- We have organized the wheels in 5 categories **(fixed, steerable, castor, Swedish and spherical)** but the last three impose NO constraint on the robot chassis. **Therefore, we only account for the fixed (f) and steerable (s) wheels.**
  - $N = N_f + Ns$,N_f = number of fixed wheels, N_s = number of steered wheels with angles φ_f, φ_s
  - Then let:

$$\varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}$$

Robot Kinematic Constraints

- **Append all Robot Rolling Constraints:**

$$J_1(\beta_s)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0$$

  - J_2 = diagonal matrix NxN whose entries are the radii r of al standard wheels
  - $J_1(\beta_s)$ = matrix with projections for all wheels to their motions along their individual wheel planes:

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}$$

$$J_2 = diag(r_1 \cdots r_N)$$

$$\varphi(t) = \underset{(N_f+N_s)\times 1}{\begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}}$$

Robot Kinematic Constraints

- **Append all Robot Rolling Constraints:**

$$J_1(\beta_s)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0$$

  - J_2 = diagonal matrix NxN whose entries are the radii r of al standard wheels
  - $J_1(\beta_s)$ = matrix with projections for all wheels to their motions along their individual wheel planes:

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}$$

$$J_2 = diag(r_1 \cdots r_N)$$

$$\varphi(t) = \underset{(N_f+N_s)\times 1}{\begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}}$$

$$\begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Robot Kinematic Constraints

- **Append all Robot Sliding Constraints:**

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}$$

  - C_{1f} , C_{1s} = diagonal matrices (Nfx3, Nsx3) whose diagonal terms are the three terms in the equations:

$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0 .$$

$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0$$

for all fixed and steering wheels correspondingly.

Robot Kinematic Constraints

- **Append all Robot Sliding Constraints:**

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix} \rightarrow \boxed{\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix}} R(\theta)\dot{\xi}_I = 0$$

  - C_{1f} , C_{1s} = diagonal matrices (Nfx3, Nsx3) whose diagonal terms are the three terms in the equations:

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix} R(\theta)\dot{\xi}_I = 0\,.$$

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix} R(\theta)\dot{\xi}_I = 0$$

for all fixed and steering wheels correspondingly.

Robot Kinematic Constraints

- **Robot Kinematic Constraints combined form:**

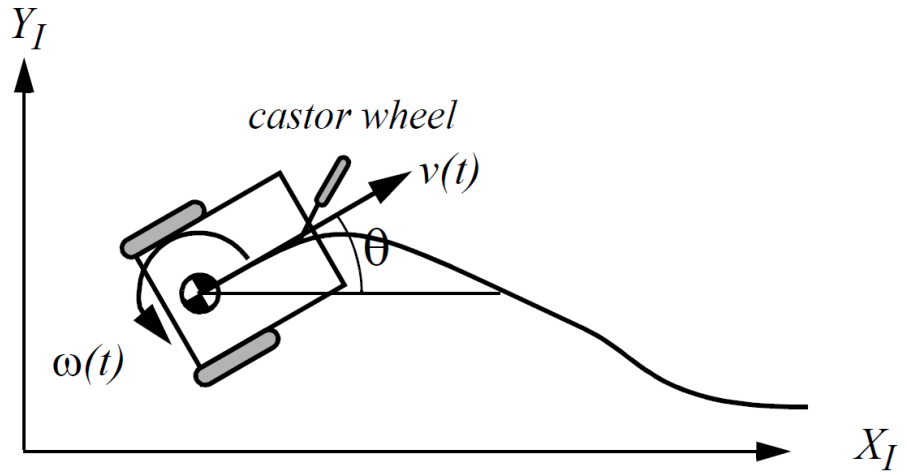
$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$

Example: Differential Drive Robot

- **Robot Kinematic Constraints combined form:**

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$

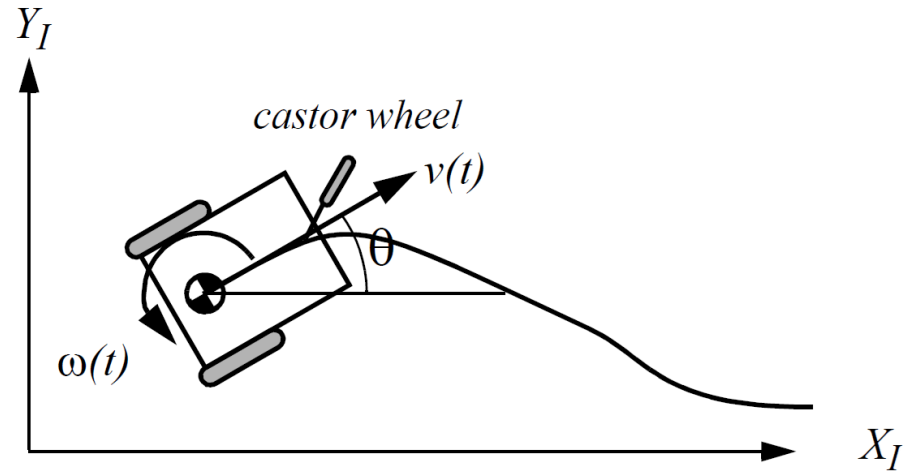
- The castor is unpowered and is free to move in any direction.
- The two remaining wheels are not steerable (so we have the fixed wheel formulas for each of them).
  - Supposing that the robot's local reference frame is aligned such that the robot moves forward along $+X_R$ then:
    - Right Wheel: $\alpha = -\pi/2, \beta = \pi$
    - Left Wheel: $\alpha = \pi/2, \beta = 0$



Example: Differential Drive Robot

- Therefore:

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$



- Through inversion:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2l} & -\frac{1}{2l} & 0 \end{bmatrix} \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$

Mobile Robot Maneuverability

- **The kinematic mobility of a robot chassis is its ability to directly move in the environment.** The basic constraint limiting mobility is the rule that every wheel must satisfy its sliding constraint. Therefore, we can formally derive robot mobility by starting from:

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

- **In addition to instantaneous kinematic motion, a mobile robot is able to further manipulate its position over time, by steering steerable wheels.** The overall maneuverability of a robot is thus a combination of the mobility available based on the kinematic sliding constraints of the standard wheels, plus the additional freedom contributed by steering and spinning the steerable standard wheels.

Degree of Maneuverability

- **Defining the constraints for fixed and steerable wheels separately:**

$$C_{1f}R(\theta)\dot{\xi}_I = 0\,.$$

$$C_{1s}(\beta_s)R(\theta)\dot{\xi}_I = 0$$

  - **For both constraints to be satisfied**, the vector $R(\theta)\dot{\xi}_1$ must belong to the null space of the projection matrix $C_1(\beta_s)$, which is a combination of C_{1f} and C_{1s}.
    - The null space of $C_1(\beta_s)$ is the space N such that for any vector $n \in N$, $C_1(\beta_s)s = 0$
- **The robot chassis kinematics is therefore a function of the set of independent constraints arising from all standard wheels.**
  - The greater the number of constraints, and therefore the greater the rank of $C_1(\beta_s)$, the more constrained is the mobility of the robot.
  - **Degree of mobility:**

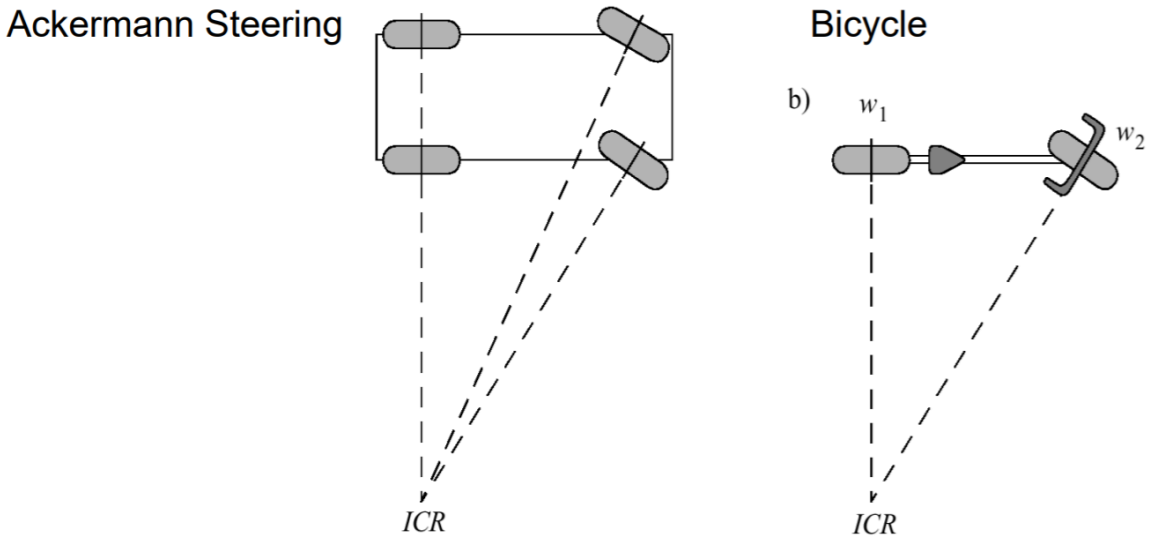
$$\delta_m = dim N\left[C_1(\beta_s)\right] = 3 - rank\left[C_1(\beta_s)\right]$$

Degree of Maneuverability

- **Degree of Mobility**

$$\delta_m = dim N\left[C_1(\beta_s)\right] = 3 - rank\left[C_1(\beta_s)\right]$$

  - Quantifies the degrees of controllable freedom based on changes to wheel velocity.



Degree of Maneuverability

- **Degree of Steerability:**

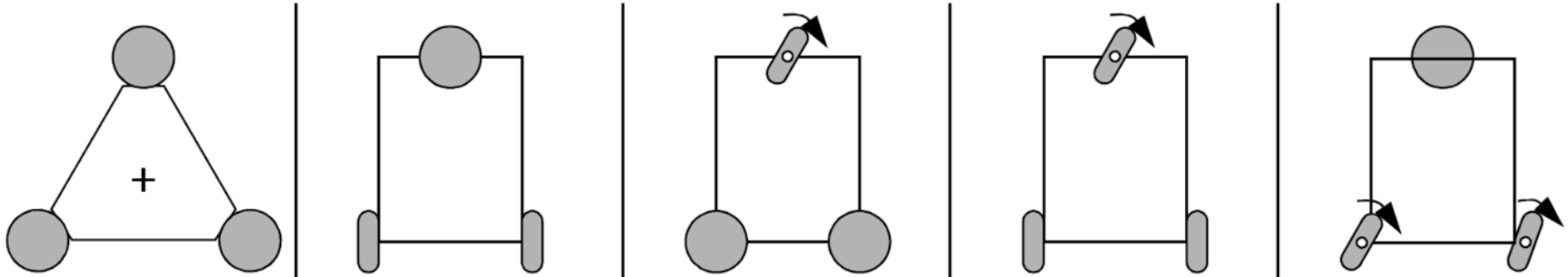
$$\delta_s = rank\left[C_{1s}(\beta_s)\right]$$

  - Quantifies the number of independently controllable steering parameters.

Degree of Maneuverability

- **Degree of Maneuverability:**

$$\delta_M = \delta_m + \delta_s$$



Omnidirectional	Differential	Omni-Steer	Tricycle	Two-Steer
δ_M =3	δ_M =2	δ_M =3	δ_M =2	δ_M =3
δ_m =3	δ_m =2	δ_m =2	δ_m =1	δ_m =1
δ_s =0	δ_s =0	δ_s =1	δ_s =1	δ_s =2

Degree of Maneuverability

- For any robot with $\delta_M = 2$ the Instantaneous Center of Rotation is always constrained to lie on a line.
- For any robot with $\delta_M = 3$ the Instantaneous Center of Rotation is not constrained and be set to any point on the plane.

Mobile Robot Workspace

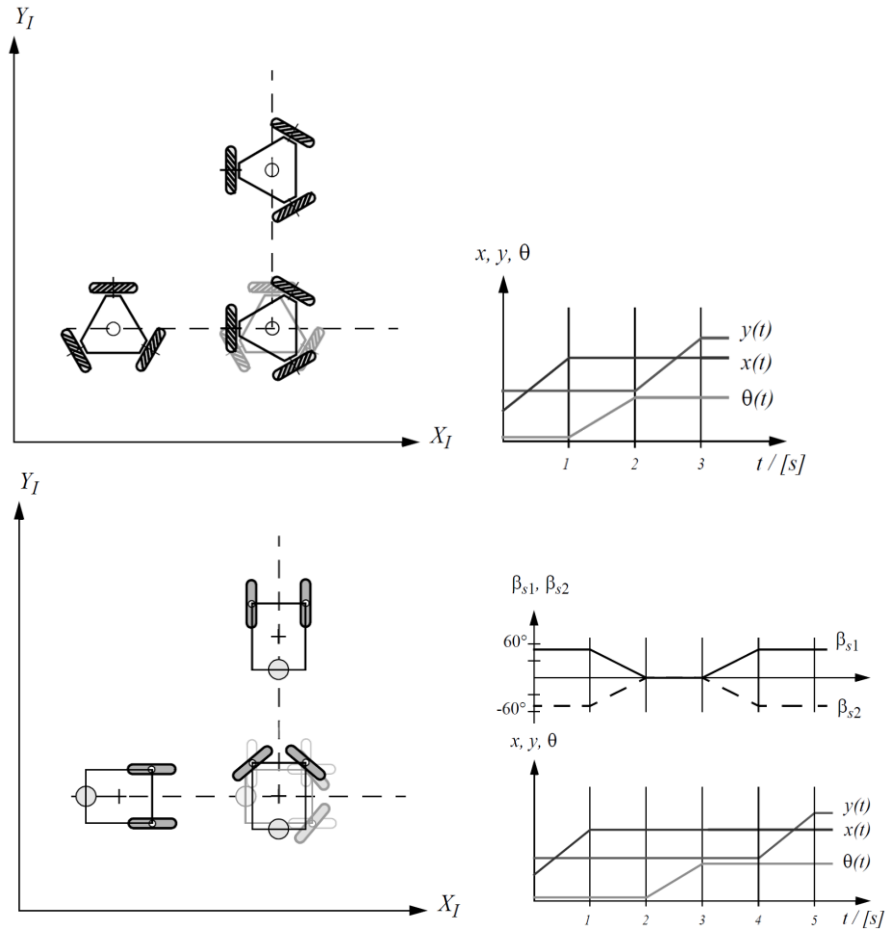
- Not all trajectories are admissible for a robot – therefore giving rise to a constraint workspace.
- *Differential Degree Of Freedom* (**DDOF**) = the number of dimensions in the velocity space of a robot is the number of independently achievable velocities or DDOF.
- A mobile robot's DDOF is equal to its degree of mobility δ_m
  - When the robot kinematic *Degrees Of Freedom* (**DOF**) are of the same number as the robot's DDOF then, this robot is **holonomic**.
    - **A holonomic robot can admit any trajectory in the collision-free space.**

Mobile Robot Workspace

- Not all trajectories are admissible for a robot – therefore giving rise to a constraint workspace.
- *Differential Degree Of Freedom* (**DDOF**) = the number of dimensions in the velocity space of a robot is the number of independently achievable velocities or DDOF.
- A mobile robot's DDOF is equal to its degree of mobility δ_m
  - When the robot kinematic *Degrees Of Freedom* (**DOF**) are of the same number as the robot's DDOF then, this robot is **holonomic**.
    - **A holonomic robot can admit any trajectory in the collision-free space.**
- Many broadly utilized robots are **nonholonomic**!
  - Main Example: Differential drive robot

Mobile Robot Workspace

- Example of holonomic and nonholonomic robot trajectories:



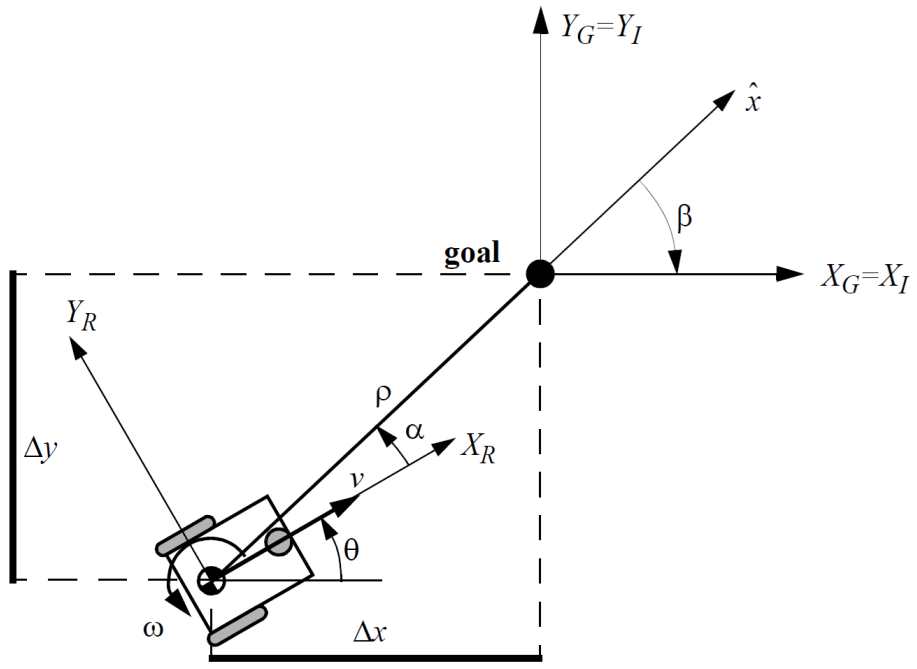
Differential Drive Robot Kinematic Model

- The kinematic model of the differential drive (nonholonomic) robot takes the form:

$$I\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$

- And:

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-\cos\alpha & 0\\ \frac{\sin\alpha}{\rho} & -1\\ -\frac{\sin\alpha}{\rho} & 0\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$



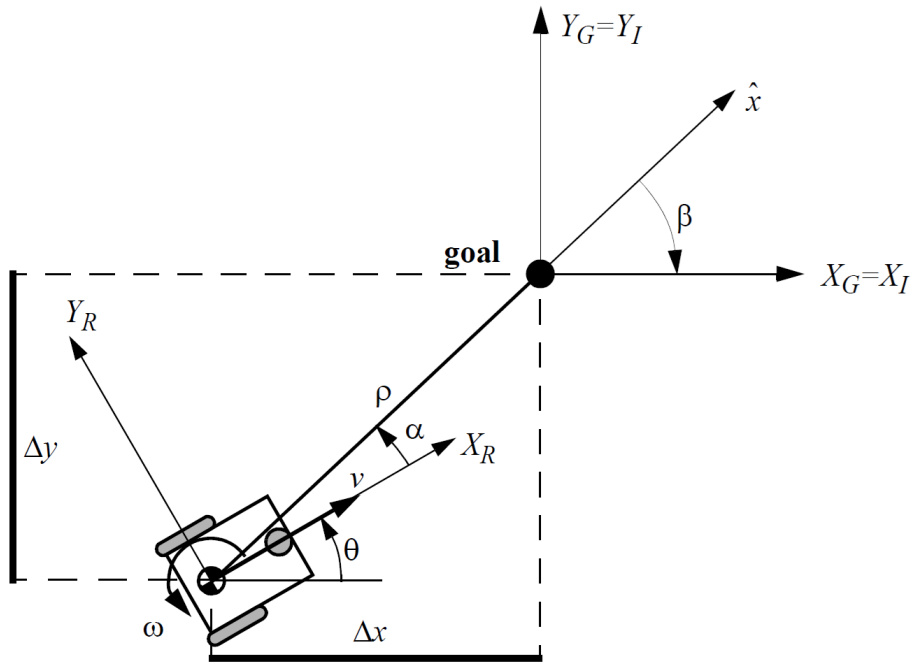
Differential Drive Robot Kinematic Model

- The kinematic model of the differential drive (nonholonomic) robot takes the form:

$$I\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$

- And:

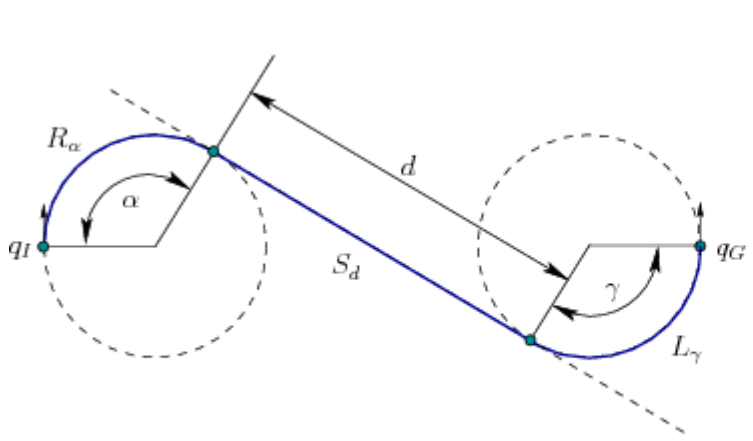
$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-\cos\alpha & 0\\ \frac{\sin\alpha}{\rho} & -1\\ -\frac{\sin\alpha}{\rho} & 0\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$



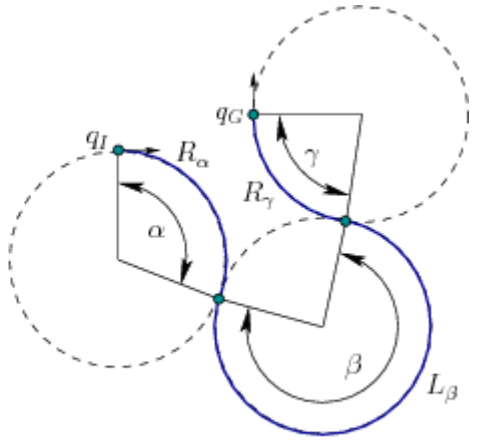
Differential Drive Robot Kinematic Model

- For differential drive robots, open-loop optimal trajectories exist: **Dubins Car** model.
- The Dubins Car model trajectories consist solely of left and right turns with maximum steering angle, and straight lines.
- The set of possible trajectories are:

{LRL, RLR, LSL, LSR, RSL, RSR}



$R_\alpha S_d L_\gamma$



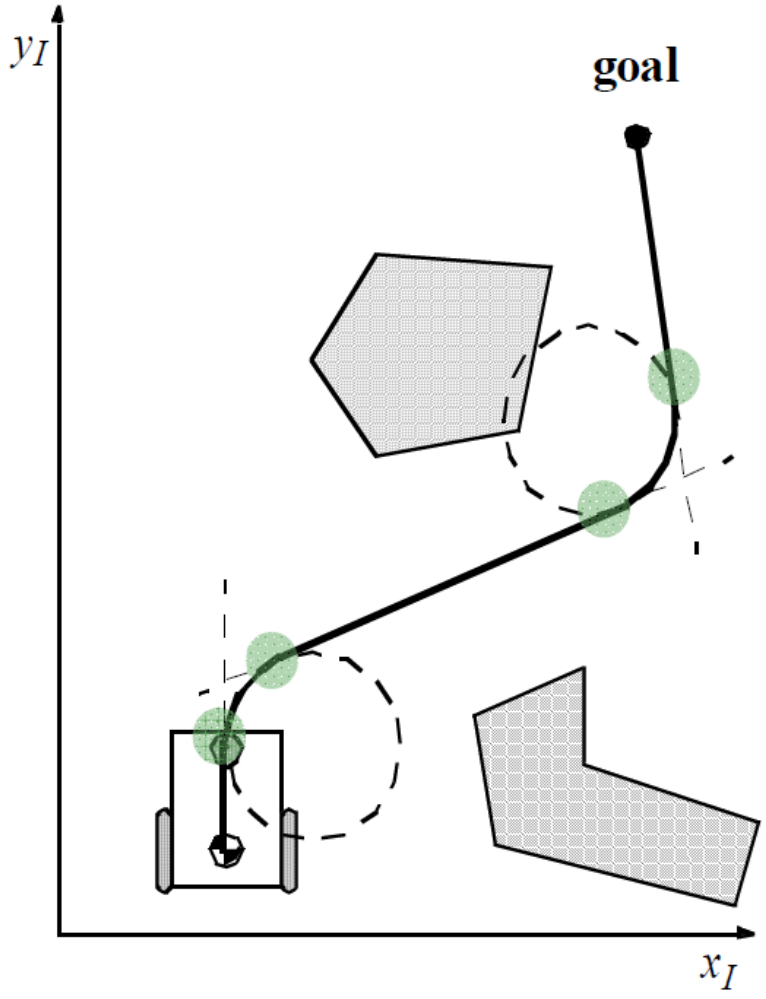
$R_\alpha L_\beta R_\gamma$

Overview

- The objective of a kinematic controller is to follow a trajectory described by its position and/or velocity profiles as function of time.
- Motion control is not straight forward because mobile robots are typically nonholonomic and MIMO systems.
- Most controllers (including the one presented here) are not considering the dynamics of the system

Motion Control: Open Loop

- Trajectory (path) divided in motion segments of clearly defined shape:
  - Straight lines and segments of a circle
  - Dubins car, and Reeds-Shepp car
- Control problem:
  - Pre-compute a smooth trajectory
    - Based on line, circle (and clothoid) segments
- Disadvantages:
  - Pre-computation of feasible trajectories is not an easy task
  - Limitations and constraints of the robots velocities and accelerations.
  - Does not adapt or correct the trajectory if dynamic changes of the environment occur.
  - The resulting trajectories are usually not smooth (ir acceleration, jerk, etc)



Motion Control: Feedback Control

- Find a control matrix K, if one exists:

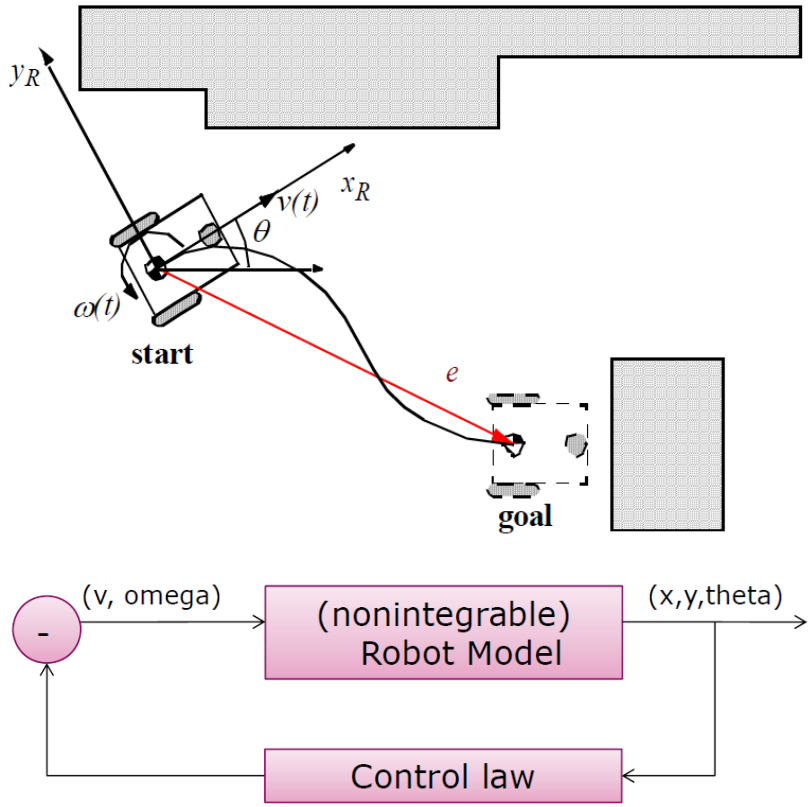
$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

  - With $k_{ij} = k(t, e)$

- such that the control of $v(t)$ and $\omega(t)$

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- drives the error e to zero: $\lim_{t \to \infty} e(t) = 0$



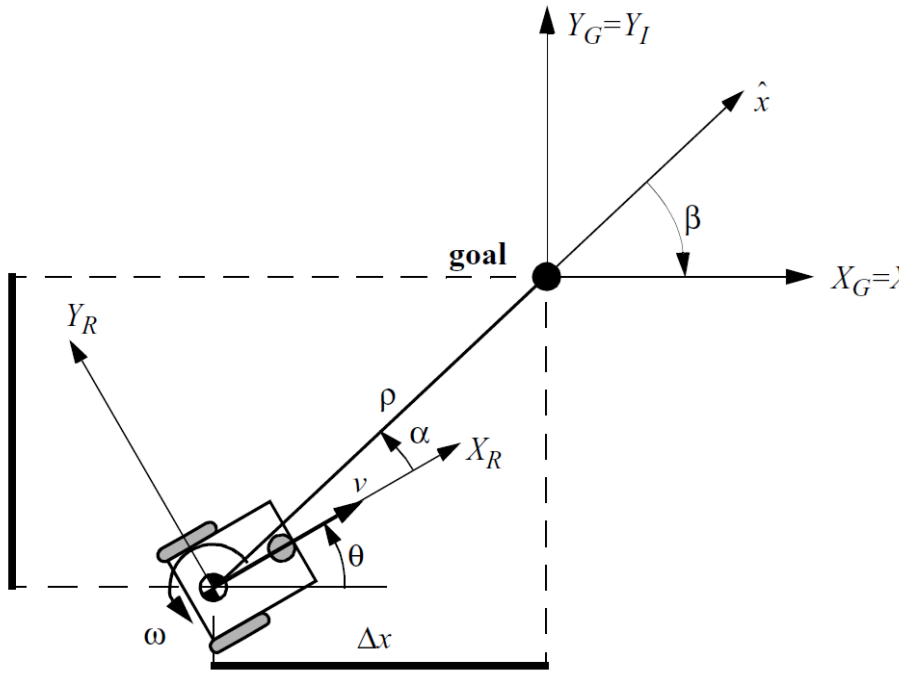
- MIMO state feedback control

Motion Control: Kinematic Position Control

- The kinematics of a differential drive mobile robot described in the inertial frame $[x_I, y_I, \theta]$ are given by:

$$^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- where $\dot{x}$ and $\dot{y}$ are the linear velocities expressed on the inertial frame.
- Let α denote the angle between the axis x_R of the robots reference frame and the vector connecting the center of the axle of the wheels with the final position.



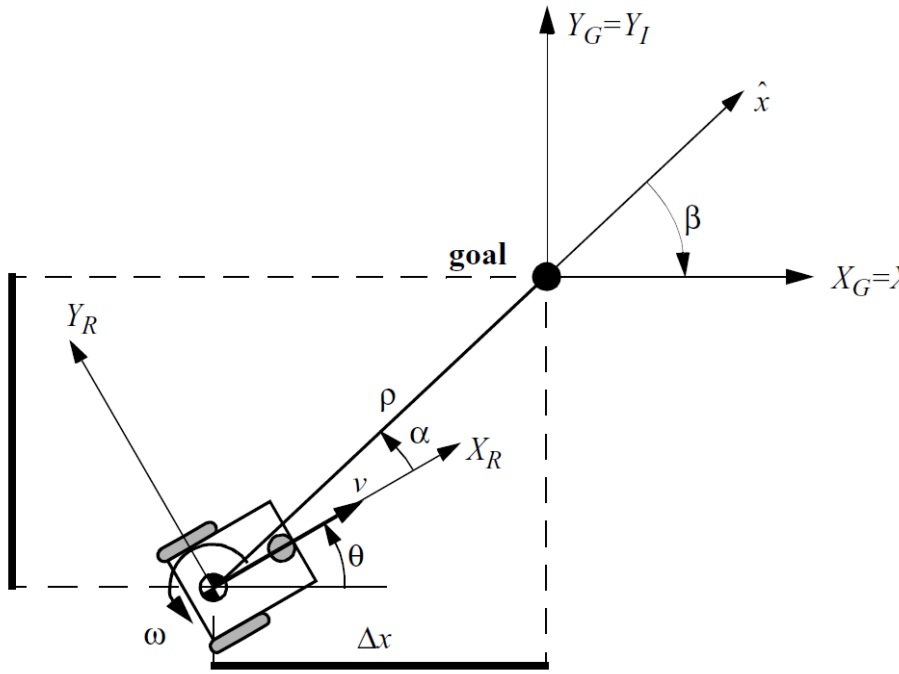
Kinematic Position Control: Coordinates Transformation

- Coordinates transformation into polar coordinates with its origin at goal position:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x)$$

$$\beta = -\theta - \alpha$$



- System description, in the new polar coordinates:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \qquad I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi]$$

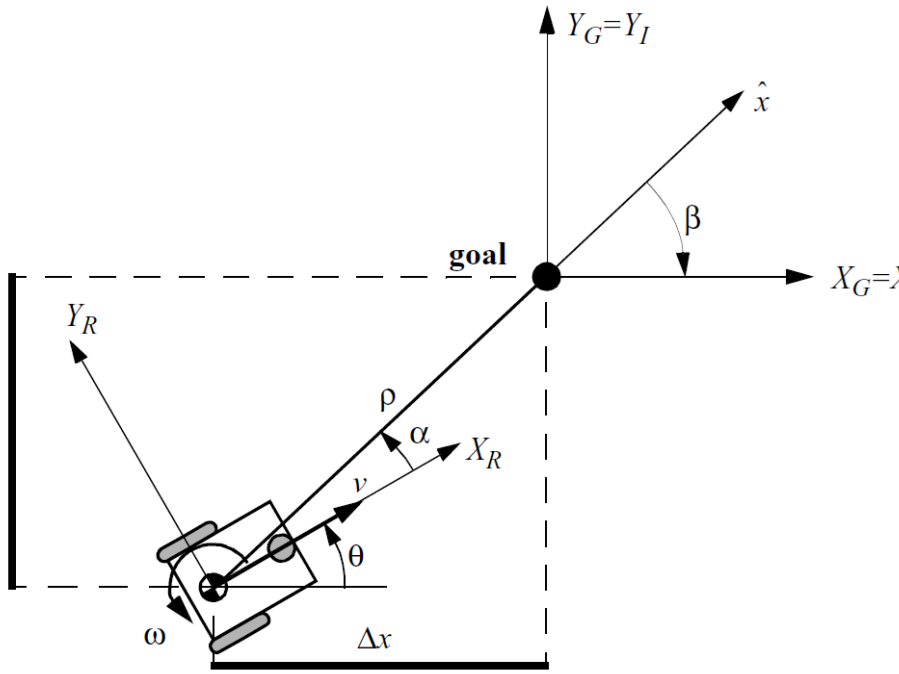
Kinematic Position Control: Coordinates Transformation

- Coordinates transformation into polar coordinates with its origin at goal position:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x)$$

$$\beta = -\theta - \alpha$$



- System description, in the new polar coordinates:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

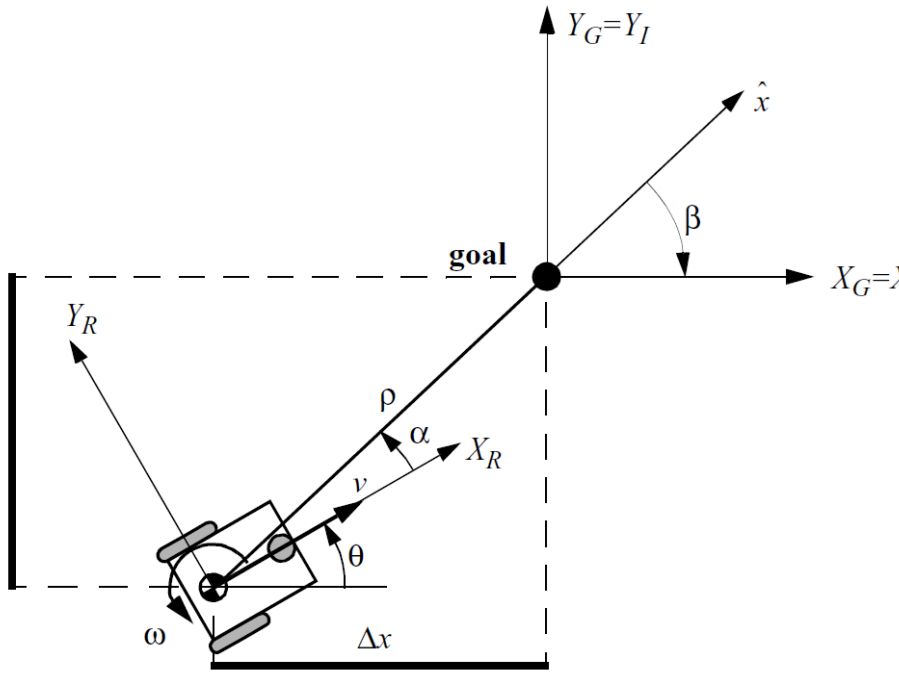
$$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \qquad I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi]$$

Kinematic Position Control: Remarks

- The coordinates transformation is not defined at $x = y = 0$
- For $a \in I_1$ the forward direction of the robot point toward the goal, for $a \in I_2$ it is the backward direction.

$$\alpha \in I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

- By properly defining the forward direction of the robot at its initial configuration, it is always possible to have $a \in I_1$ at $t = 0$. However, this does not mean that a remains in I_1 for all time t.



Kinematic Position Control: The Control Law

- It can be shown that with:

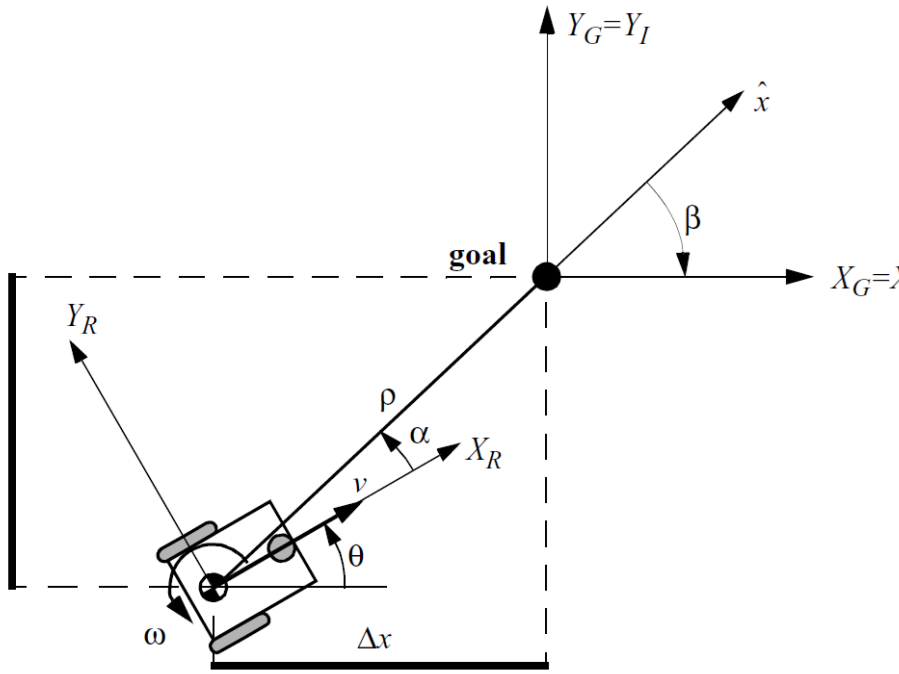
$$v = k_\rho \rho$$

$$\omega = k_\alpha \alpha + k_\beta \beta$$

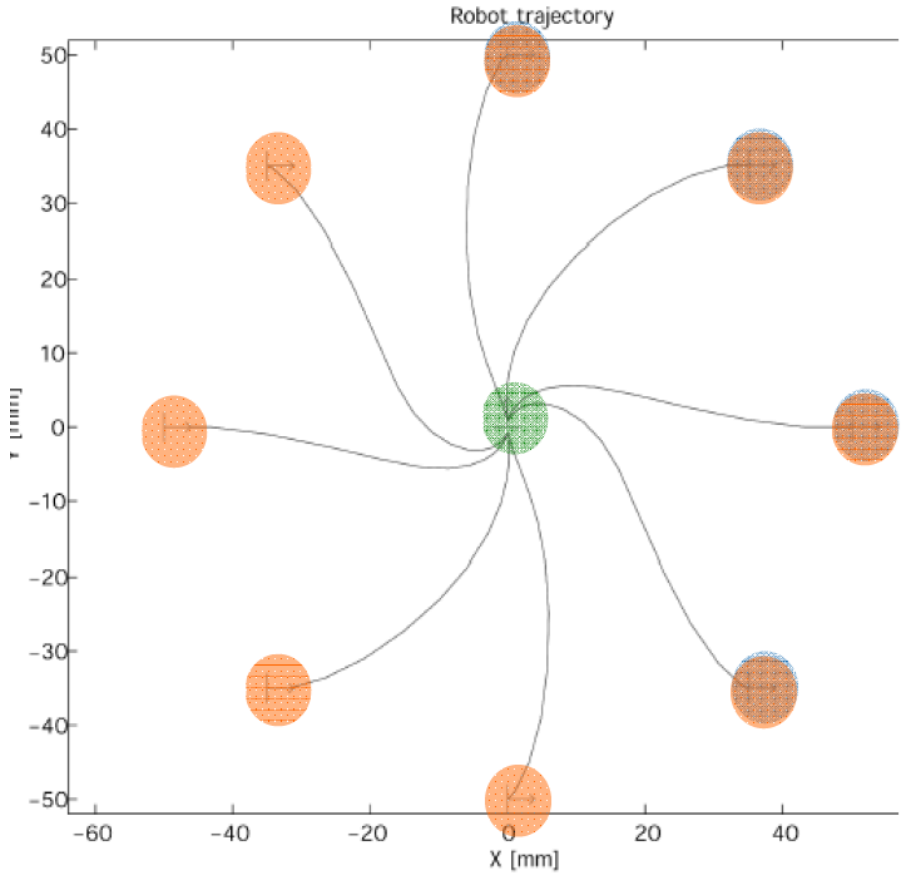
- The feedback control system

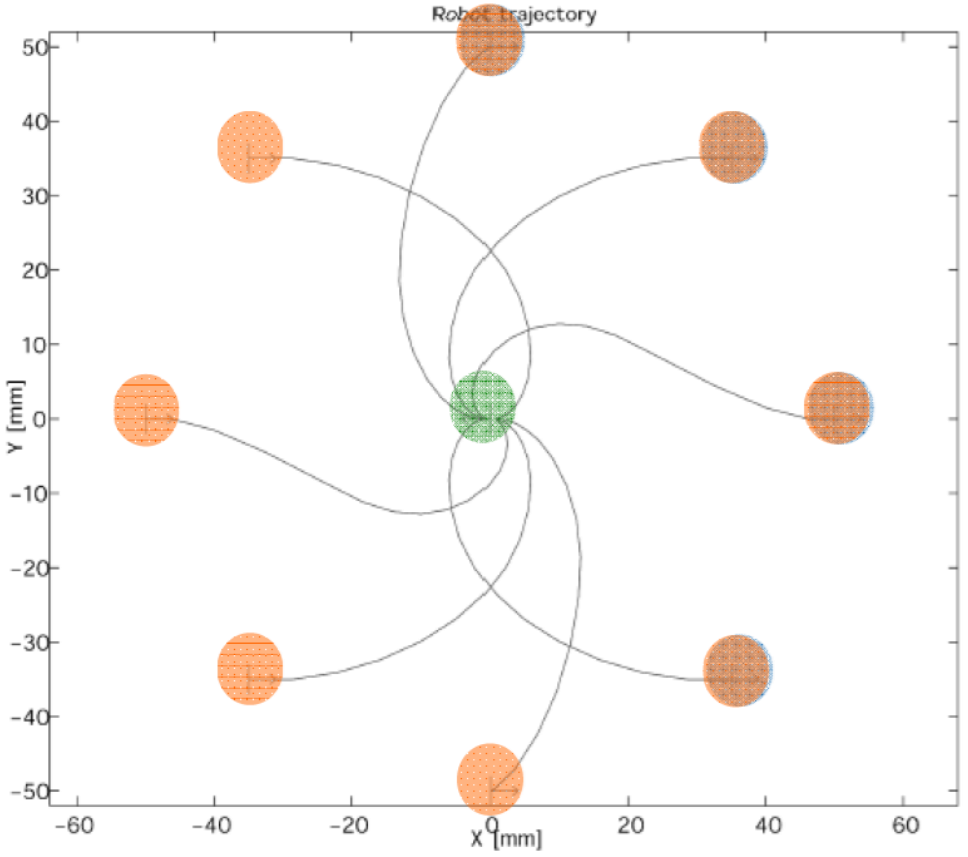
$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin\alpha \end{bmatrix}$$

- Will drive the robot to $[\rho, \alpha, \beta] = [0,0,0]$
- The control signal v has always constant sign
  - The direction of movement is kept positive or negative during the movement
  - A parking maneuver is always performed in the most natural way, and without even inverting its motion.



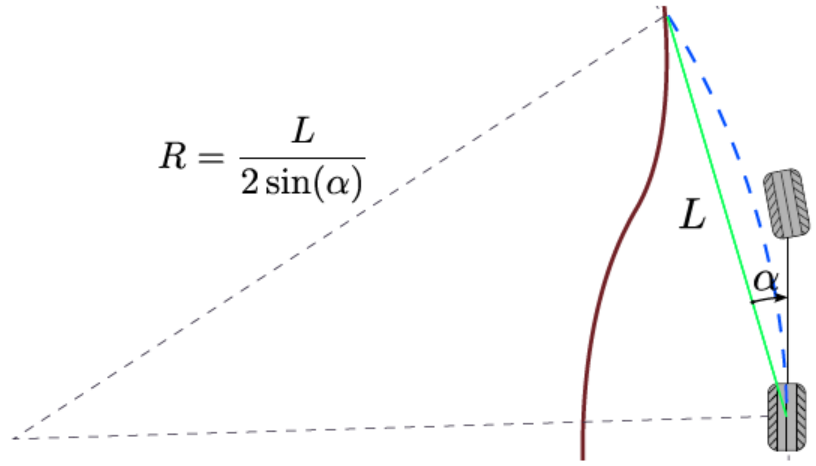
Kinematic Position Control: Resulting Path





Pure Pursuit Control

- Among the earliest proposed path tracking strategies.
- Despite its simplicity, it has been an integral part of some of the most successful projects including the DARPA Grand Challenge and the DARPA Urban Challenge.
- The control is based on fitting a semi-circle through the vehicle's current configuration to a point on the reference path ahead of the vehicle by a distance L



Pure Pursuit Control

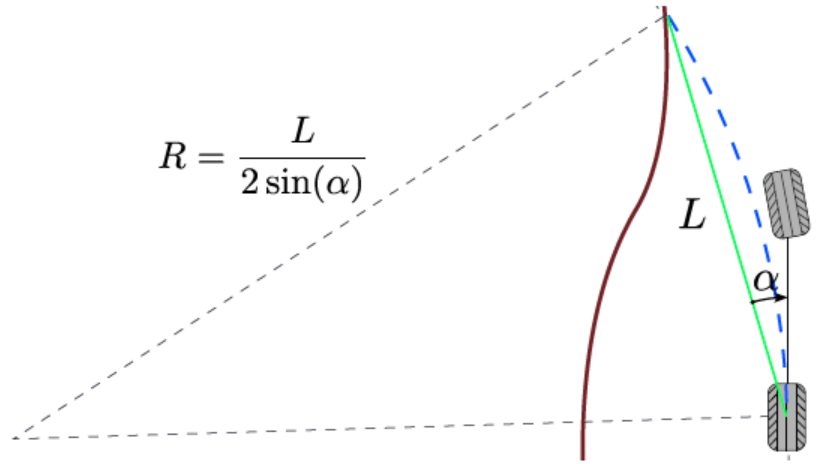
- The circle is defined as passing through the position of the car and the point on the path ahead of the car by one lookahead distance with the circle tangent to the car's heading.
- The curvature of the circle is given by:

$$\kappa = \frac{2\sin(\alpha)}{L}$$

- For a vehicle speed v_r, the commanded heading rate is

$$\omega = \frac{2v_r \sin(\alpha)}{L}$$

- Angle a can be expressed in terms of the inertial coordinate system to define a state feedback control.



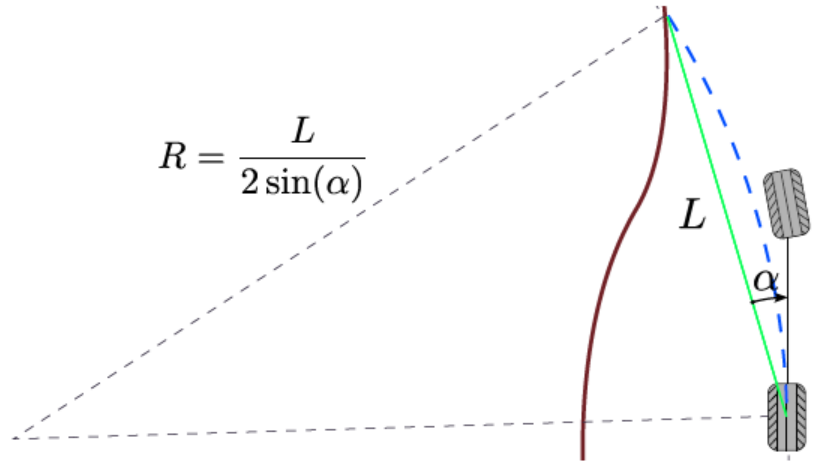
Pure Pursuit Control

- Consider the configuration $(x_r, yr, \theta)^T$ and the points on the path $(x_{ref}(s), y_{ref}(s))$, such that $\left\|\left(x_{ref}(s), y_{ref}(s)\right) - (x_r, y_r)\right\| = L$
- Since there is generally more than one such point on the reference, take the one with the greatest value of the parameter Type equation here. s to uniquely define a control law.

$$\alpha = \arctan\left(\frac{y_{ref} - y_r}{x_{ref} - x_r}\right) - \theta$$

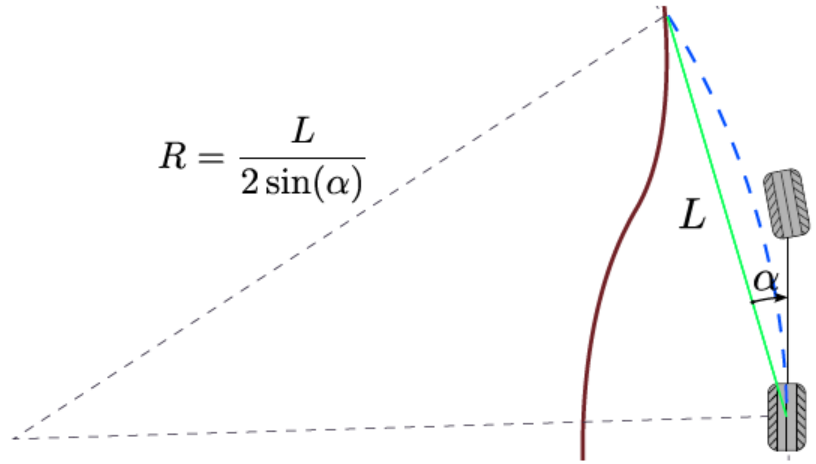
- For a path that has no curvature and the vehicle speed is constant (even if negative), the pure pursuit controller solved the stabilization problem:

$$\begin{aligned}
&1) \quad \|x(t_1) - x_{ref}(s(t_1))\| \leq \delta \\
&\quad \Rightarrow \|x(t_2) - x_{ref}(s(t_2))\| \leq \varepsilon \\
&2) \ \lim_{t\to\infty} \|x(t) - x_{ref}(s(t))\| = 0 \\
&3) \ \lim_{t\to\infty} \dot{s}(t) = v_{ref}(s(t)).
\end{aligned}$$

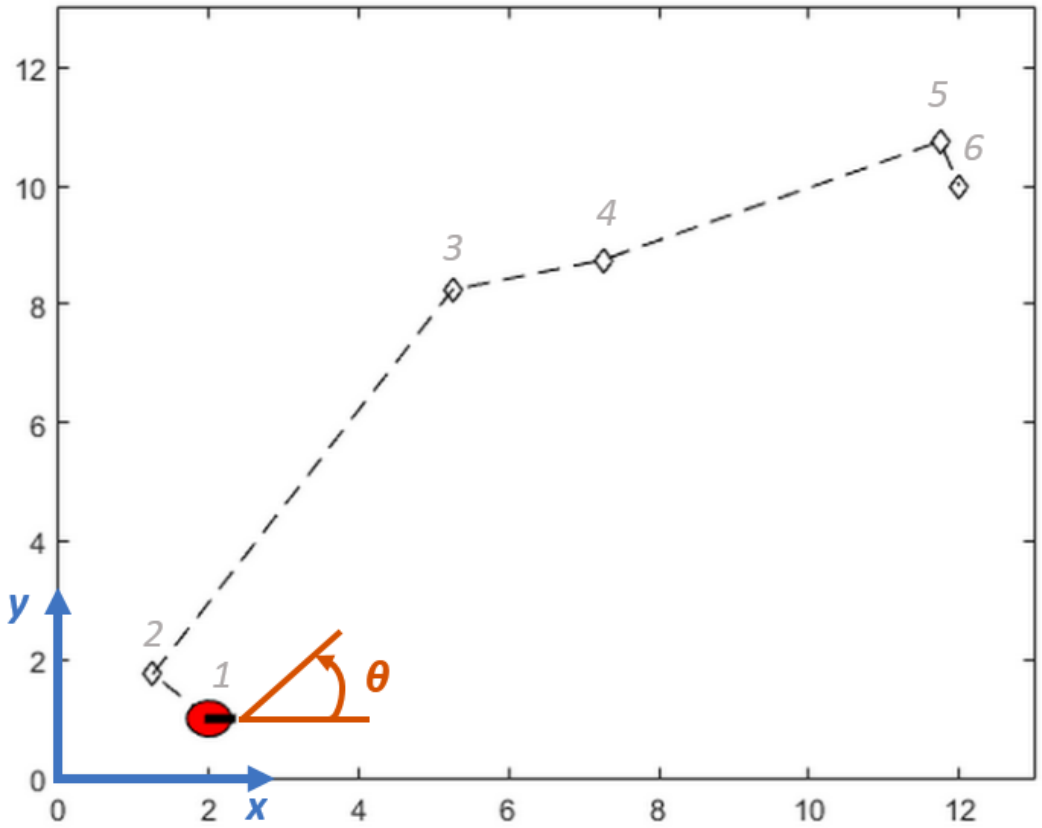


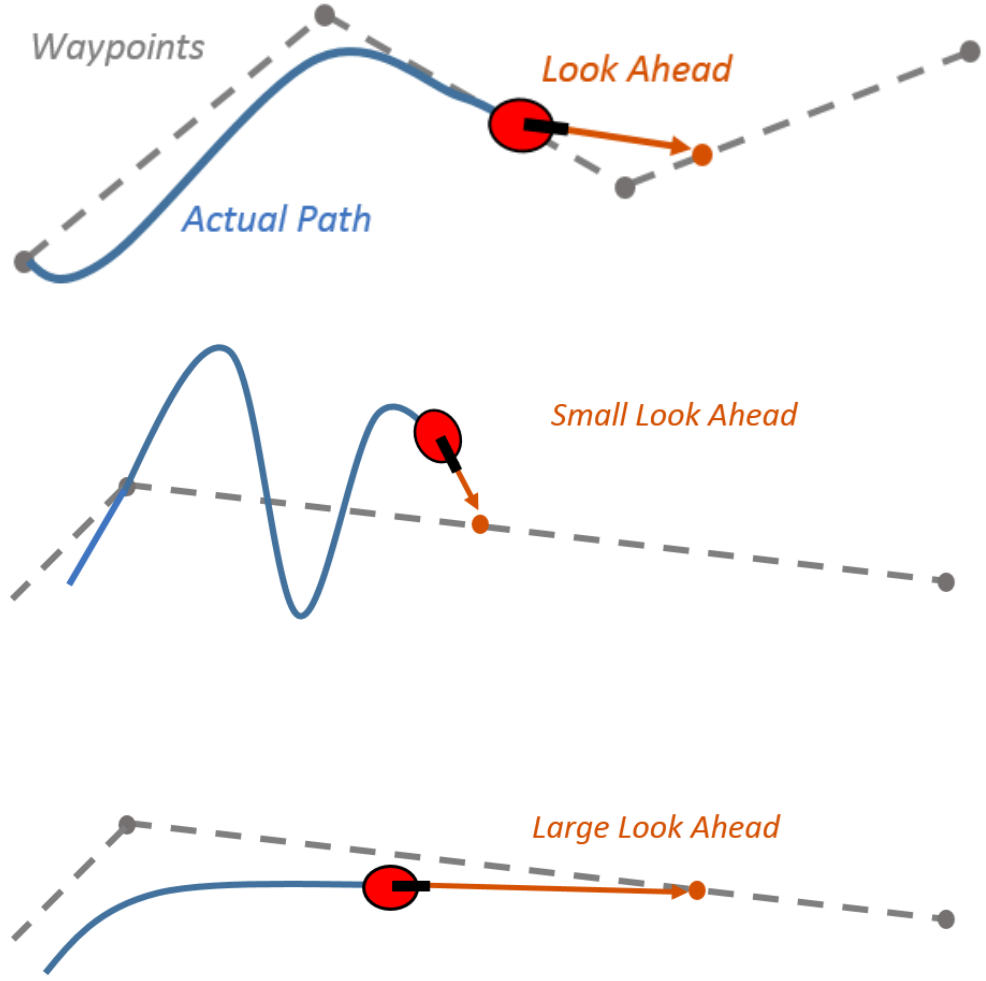
Pure Pursuit Control

- For a fixed nonzero curvature, the pure pursuit controller has a small steady state tracking error.
- In the case where the vehicle's distance to the path is greater than L, the controller output is not defined.
- Changes in reference path curvature can lead to the car deviating from the reference trajectory.
- The heading rate ω becomes increasingly sensitive to the feedback angle a as the vehicle speed increases.
  - Typically we scale L with speed to solve that.



Pure Pursuit Control





L1 Guidance Law

- The L1 Guidance law enables following of curved or straight trajectories from a fixed-wing UAV.
- It uses the UAV's velocity, as well as its change in velocity due to "external disturbances" such as wind.
- It tracks its instantaneous speed to quickly adapt to velocity changes.

Atlantik Solar

Test Flight Day #24
(May 18th 2015)

Aircraft: AtlantikSolar UAV Prototype (AS-P)
Location: Rothenthurm, Switzerland
Flights performed: 3
Mission: High-fidelity 3-D mapping of area using pre-computed paths with guaranteed coverage

L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV's reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:

$$a_c = \frac{v^2}{r}$$

- Where v is the velocity of the UAV, and r is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration $a_{s,cmd}$takes the form:

$$a_{s,cmd} = \frac{2v^2}{L1} sin(\eta)$$

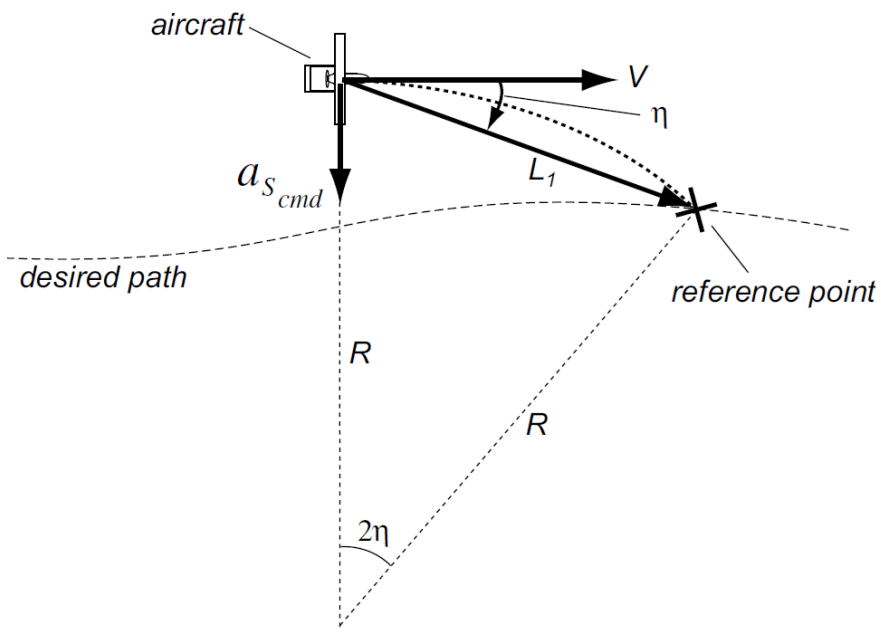
L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV's reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:

$$a_c = \frac{v^2}{r}$$

- Where v is the velocity of the UAV, and r is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration $a_{s,cmd}$ takes the form:

$$a_{s,cmd} = \frac{2v^2}{L1} sin(\eta)$$



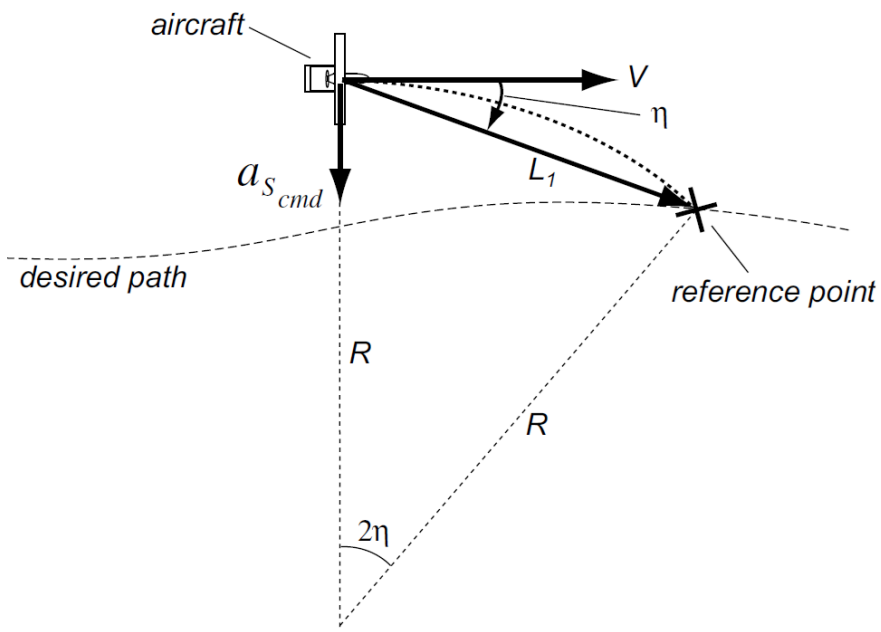
L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV's reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:

$$a_c = \frac{v^2}{r}$$

- Where v is the velocity of the UAV, and r is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration $a_{s,cmd}$ takes the form:

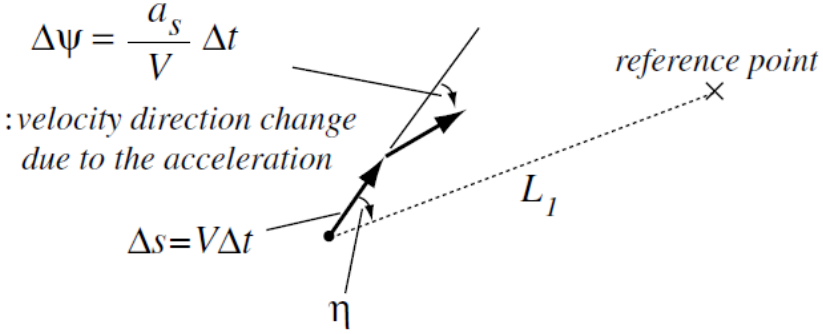
$$a_{s,cmd} = \frac{2v^2}{L1} sin(\eta)$$

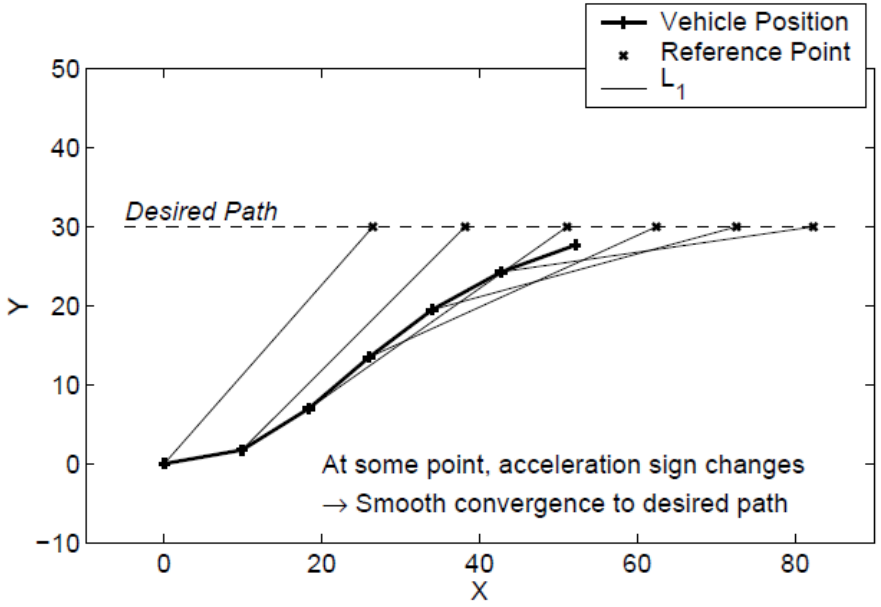


Knowing the direction of the L1 vector, the UAV will try to align its velocity vector with the L1 vector.

L1 Guidance Law

- Knowing the direction of the L1 vector, the UAV will try to align its velocity vector with the L1 vector.
- If the current path of the UAV is far off from its planned trajectory, the guidance logic turns the UAV a great amount to assist it to come back on track.





L1 Guidance: Tracking straight lines

- Centripetal acceleration required to bring the UAV on track is approximated by:

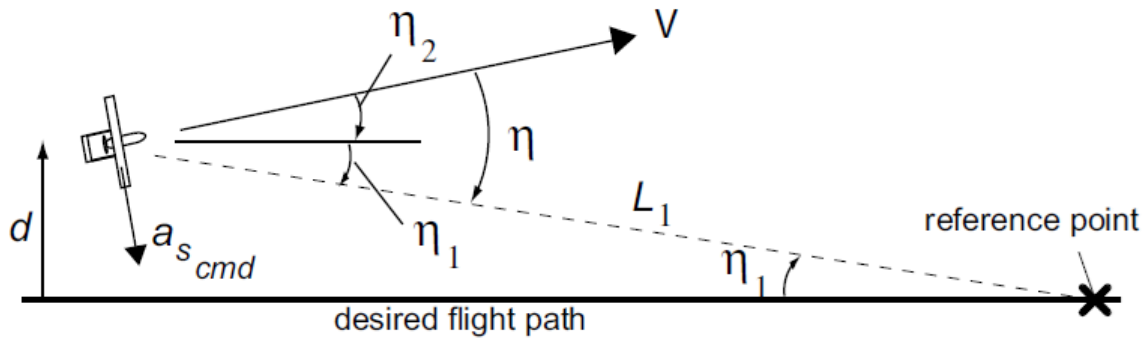
$$sin(\eta) \approx \eta$$

- As η is considered to be small, so are η_1 and η_2. $\sin\eta \approx \eta = \eta_1 + \eta_2$
- Let d denote the cross-track error – the distance perpendicular to the desired flight path to the UAV. Then:

$$\eta_1 \approx sin(\eta_1) = \frac{d}{L1} \qquad \eta_2 \approx sin(\eta_2) = \frac{\dot{d}}{V}$$

- Combining the above with the guidance formula:

$$a_{s_{cmd}} = 2\frac{V^2}{L_1}\sin\eta \approx 2\frac{V}{L_1}\left(\dot{d} + \frac{V}{L_1}d\right)$$



L1 Guidance: Tracking straight lines

- Let us also assume no-inner-loop dynamics. Then as long as η_2 is small:

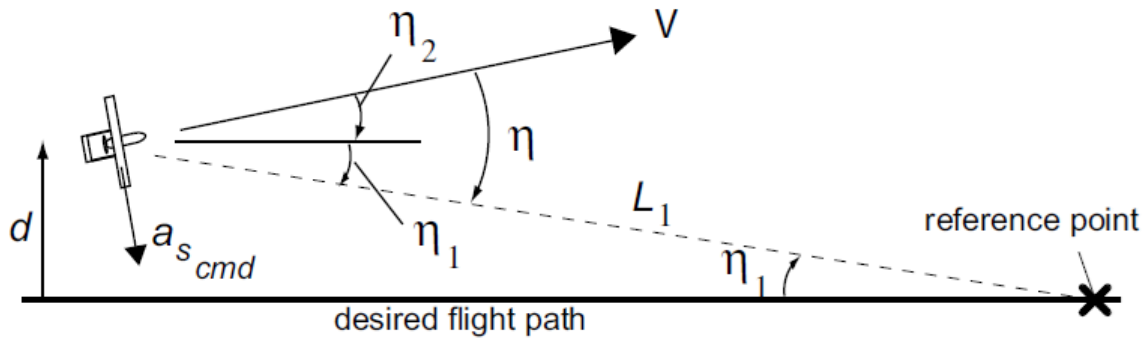
$$a_{s_{cmd}} \approx -\ddot{d}$$

- Subsequently equation:

$$a_{s_{cmd}} = 2\frac{V^2}{L_1}\sin\eta \approx 2\frac{V}{L_1}\left(\dot{d} + \frac{V}{L_1}d\right)$$

- Can be reformulated as:

$$\ddot{d} + 2\zeta\omega_n\dot{d} + \omega_n^2 d = 0 \qquad \zeta = 1/\sqrt{2},\ \omega_n = \sqrt{2}V/L_1$$



L1 Guidance: Tracking non-straight lines

- Assuming again small angles-analysis:

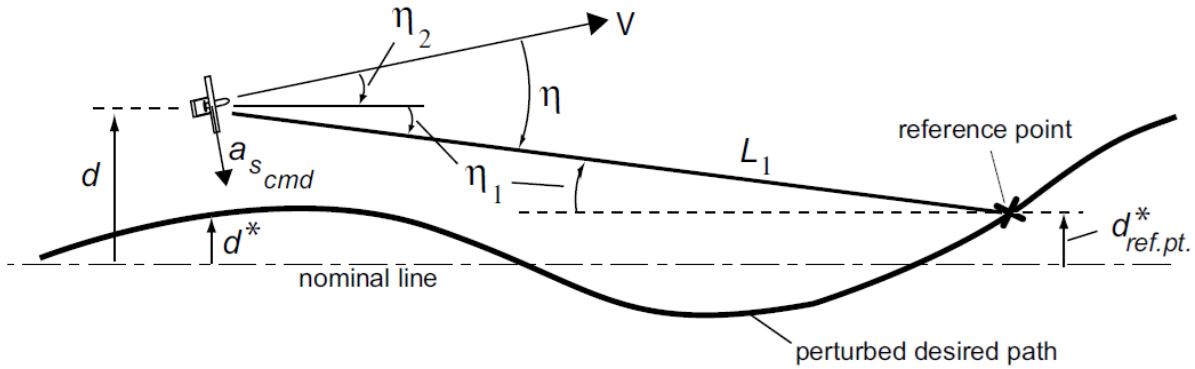
$$\sin\eta \approx \eta = \eta_1 + \eta_2$$

- Then since:

$$\eta_1 \approx \frac{d - d^{\star}_{ref.pt.}}{L_1}, \qquad \eta_2 \approx \frac{\dot{d}}{V}$$

- And $a_{s_{cmd}} \approx -\ddot{d}$, the guidance law reduces to:

$$\ddot{d} + \frac{2V}{L_1}\dot{d} + \frac{2V^2}{L_1^2}d = \frac{2V^2}{L_1^2}d^{\star}_{ref.pt.}$$



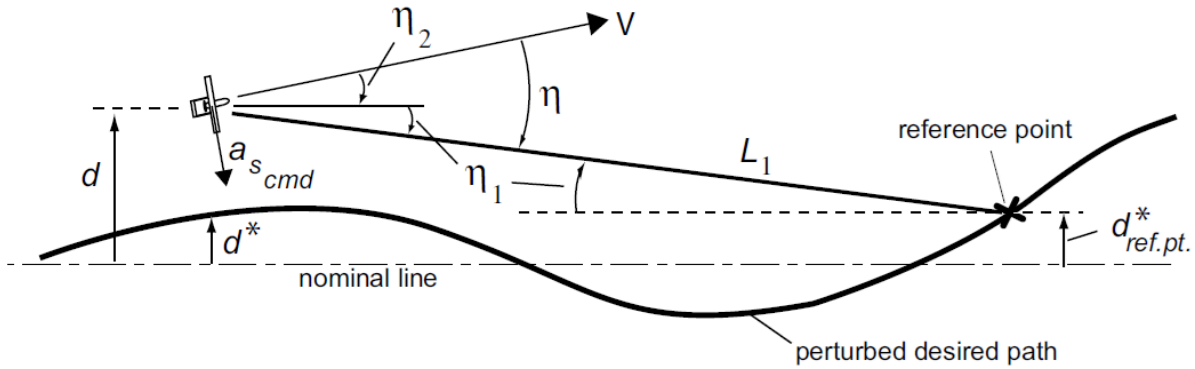
L1 Guidance: Tracking non-straight lines

- In general, considering the vehicle speed and the length for L1, and assuming small angle for η there is a time difference of approximately $L1/V$ between d^* and $d^*_{ref,pt.}$

$$\frac{d^\star_{ref.pt}(s)}{d^\star(s)} \approx e^{\tau s}, \qquad \tau \approx L_1/V$$

- Therefore:

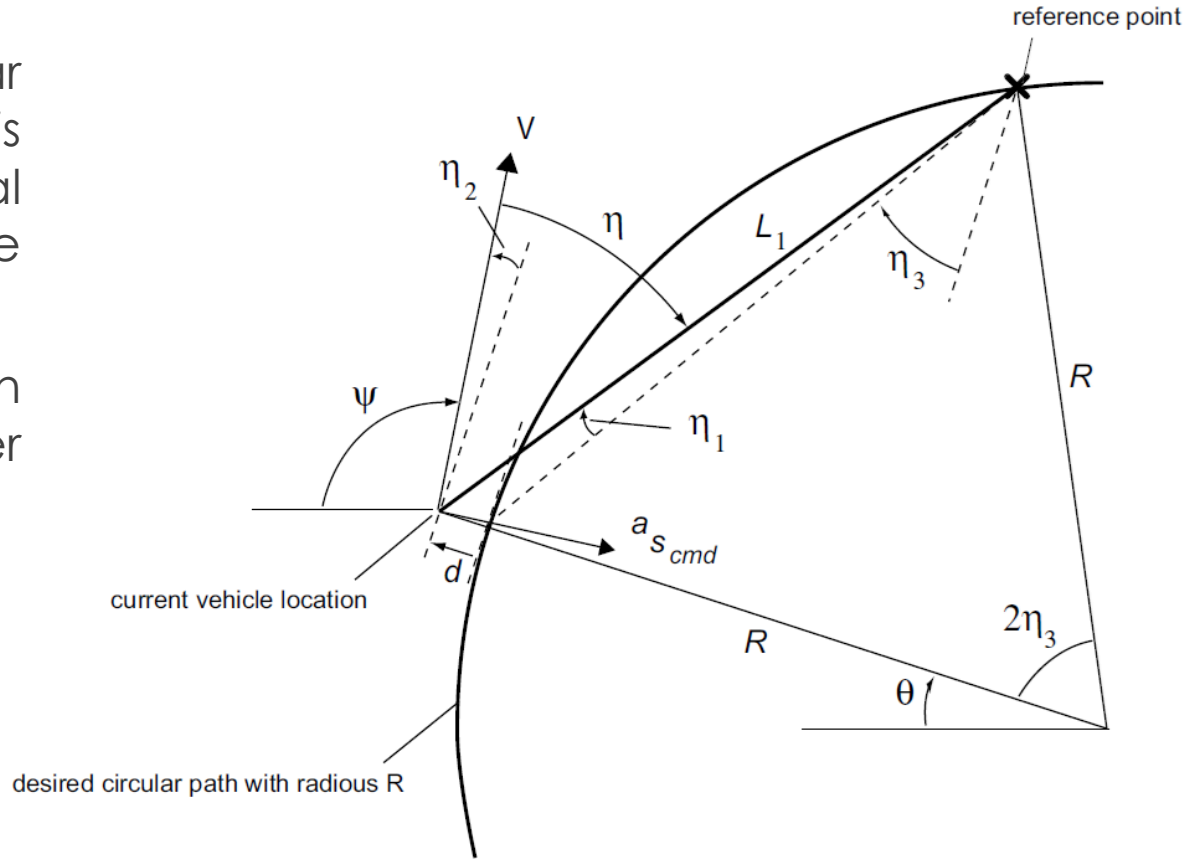
$$\frac{d(s)}{d^\star(s)} = \frac{\omega_n^2 e^{\tau s}}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad \text{where } \zeta = 0.707,\ \omega_n = \frac{\sqrt{2}V}{L_1},\ \tau \approx L_1/V$$



L1 Guidance: Tracking circular paths

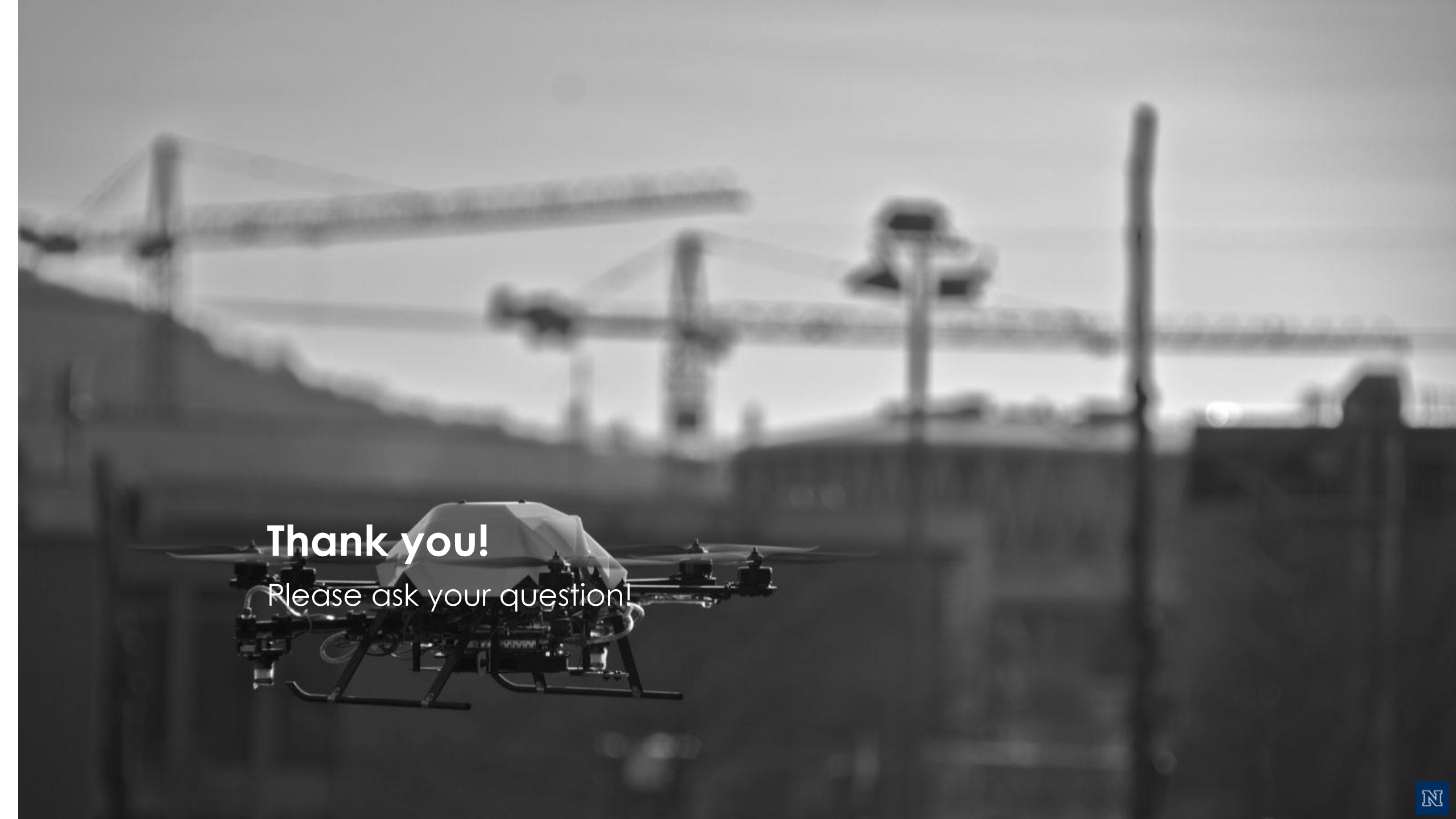
- When a UAV is following a circular path, the centripetal acceleration is aiming to be normal to the tangential acceleration it would have if it were following the circle perfectly.
- With simplifications, the circular path model also produces a second order linear system:

$$0 \approx \ddot{d} + 2\zeta\omega_n\dot{d} + \omega_n^2 d$$



Test Flight Days #9&#10
(May 5th & 6th 2014)

Aircraft: AtlantikSolar UAV Prototype
Location: Tuggen, Switzerland
Flights performed: 5
Tests: Autopilot Waypoint-following



Thank you!

Please ask your question!