CADi
ROS as a Development Platform
Kinematic Lateral Control

Renato Galluzzi
renato.galluzzi@tec.mx

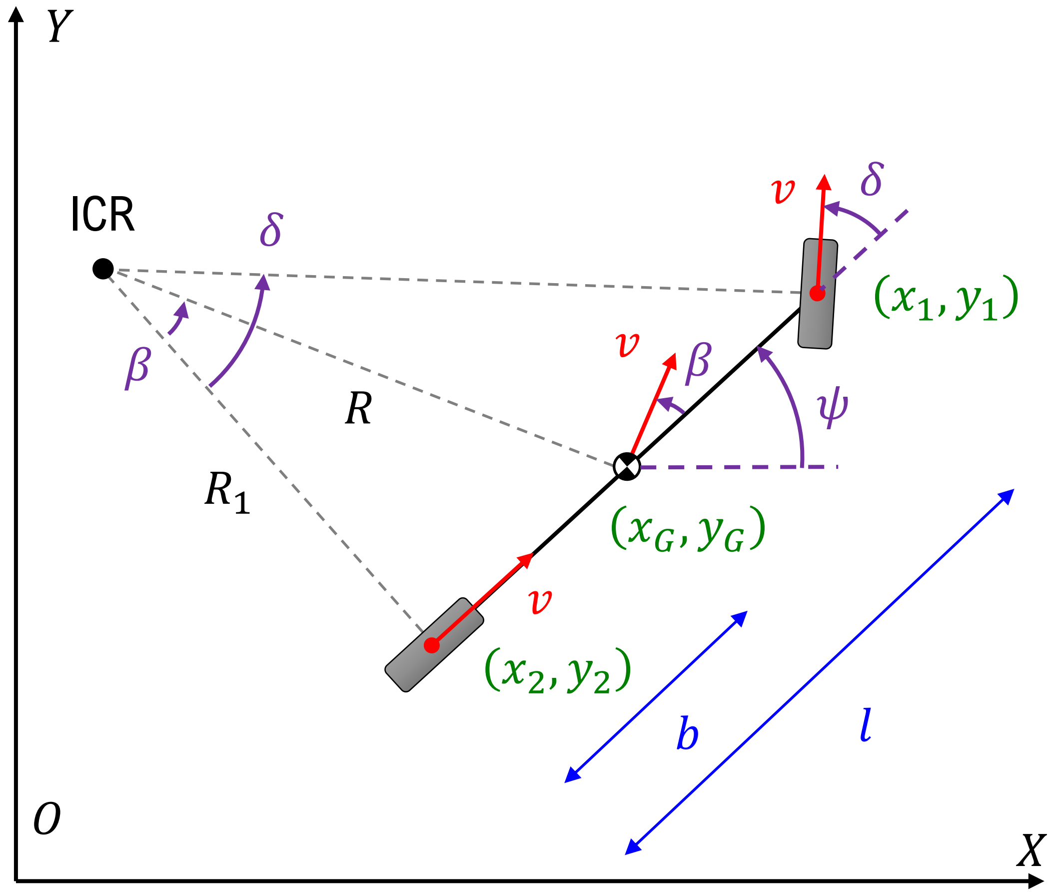
July 2020

Types of Lateral Control

- **Kinematic or geometric control:**
  - It exploits geometric relationships between the vehicle and the path.
  - Simple and effective control laws.
  - Could potentially fall in the presence of heavy dynamic content.
  - **Examples:** pure pursuit (carrot following) and Stanley.

- **Dynamic control:**
  - It considers the nonlinear model of lateral dynamics.
  - More accurate results than the kinematic variant.
  - Computationally expensive.
  - **Examples:** model predictive control, feedback linearization, sliding mode control.

Kinematic Bicycle Model



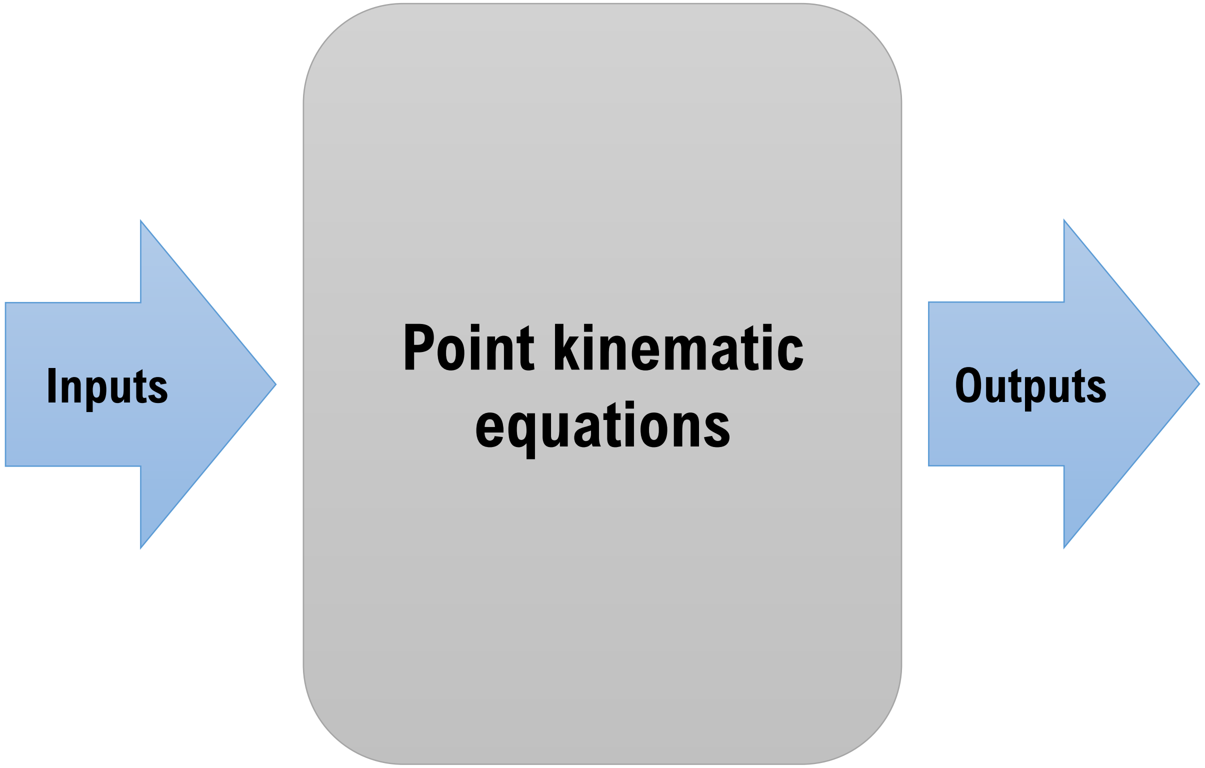
- **Motion in three possible points of interest:**
  - **rear axle** (x_2, y_2)
  - **front axle** (x_1, y_1)
  - **center of gravity (COG)** (x_G, y_G)
- We determine the instantaneous center of rotation (ICR).
- All points experience the same longitudinal speed v.

*Vehicle geometry denoted by wheelbase l and distance between rear axle and COG b

Approach

Longitudinal speed v

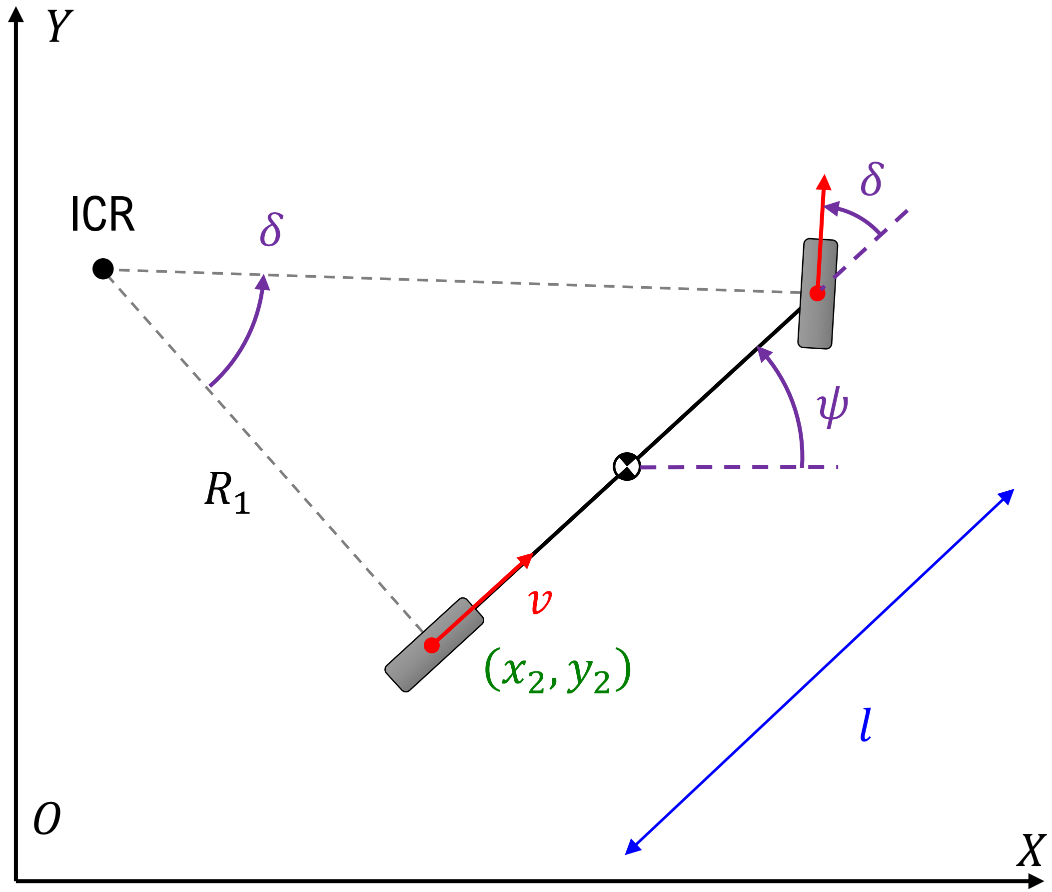
Steering angle δ



Coordinates x, y

Yaw angle ψ

Rear Axle Kinematics



$$\dot{x}_2 = v\cos\psi$$

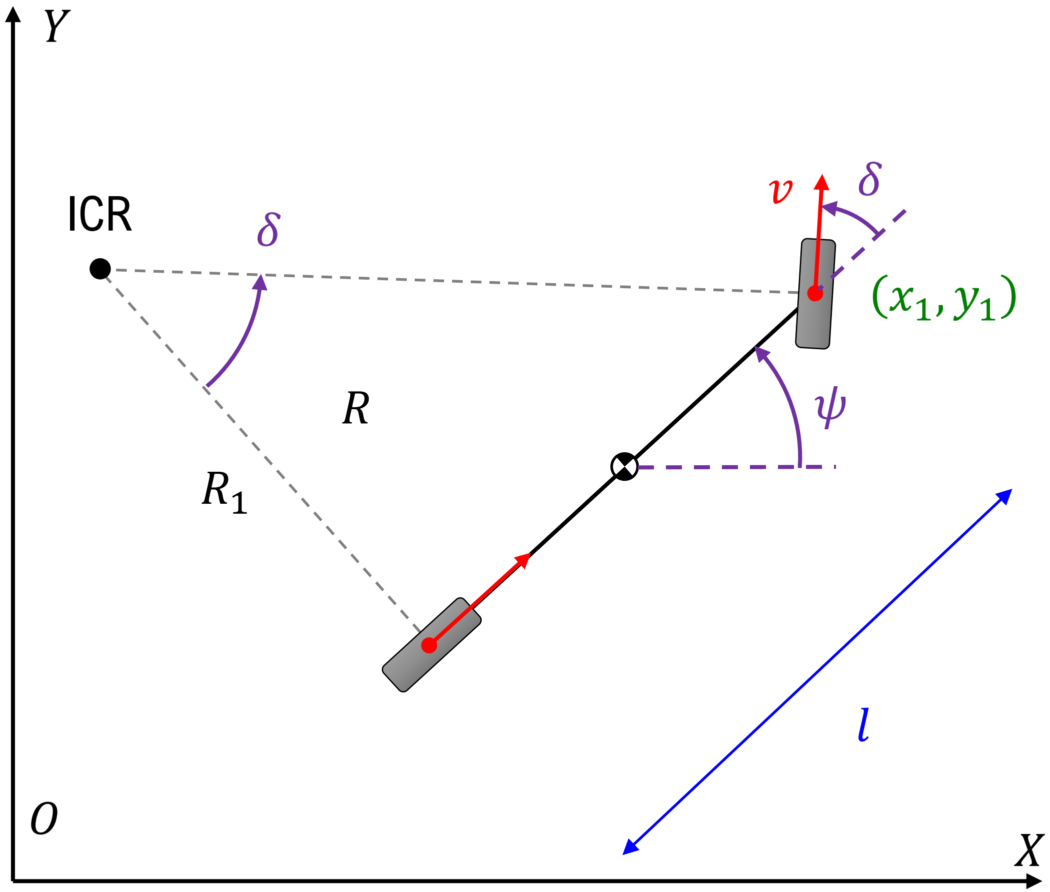
$$\dot{y}_2 = v\sin\psi$$

$$\dot{\psi} = \frac{v}{R_1}$$

$$\tan\delta = \frac{l}{R_1}$$

$$\dot{\psi} = \frac{v\tan\delta}{l}$$

Front Axle Kinematics



$$\dot{x}_1 = v\cos(\psi + \delta)$$

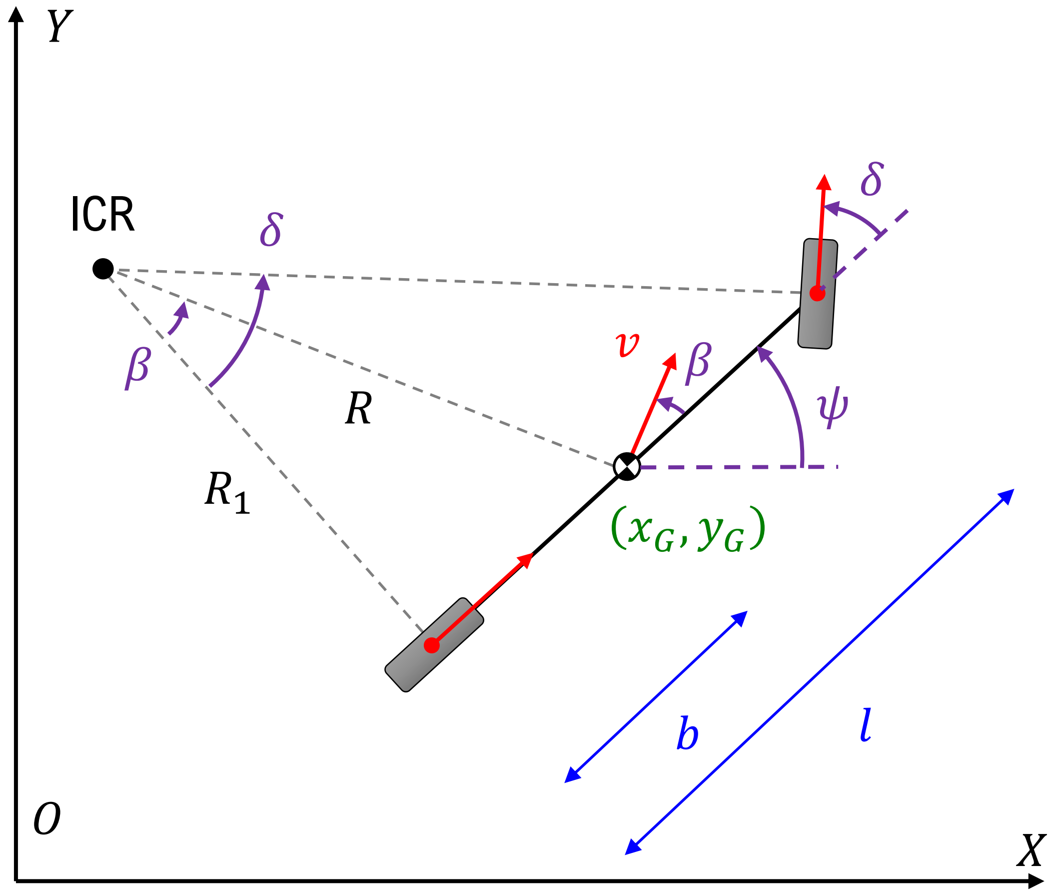
$$\dot{y}_1 = v\sin(\psi + \delta)$$

$$\dot{\psi} = \frac{v\cos\delta}{R_1}$$

$$\tan\delta = \frac{l}{R_1}$$

$$\dot{\psi} = \frac{v\cos\delta\tan\delta}{l} = \frac{v\sin\delta}{l}$$

COG Kinematics



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

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

$$\dot{\psi} = \frac{v\cos\beta}{R_1}$$

$$\tan\delta = \frac{l}{R_1}$$

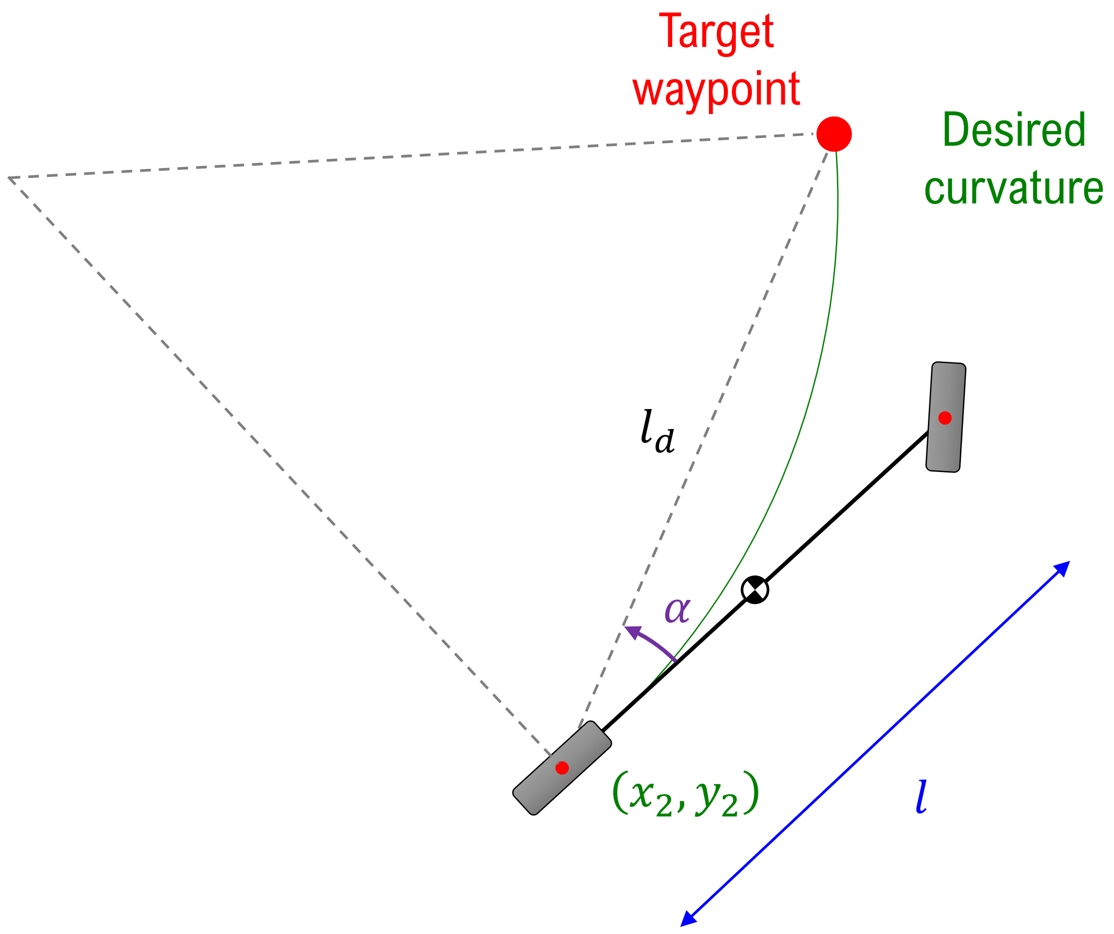
$$\dot{\psi} = \frac{v\cos\beta\tan\delta}{l}$$

$$\beta = \arctan\left(\frac{b}{R_1}\right) = \arctan\left(\frac{b\tan\delta}{l}\right)$$

Pure Pursuit Control

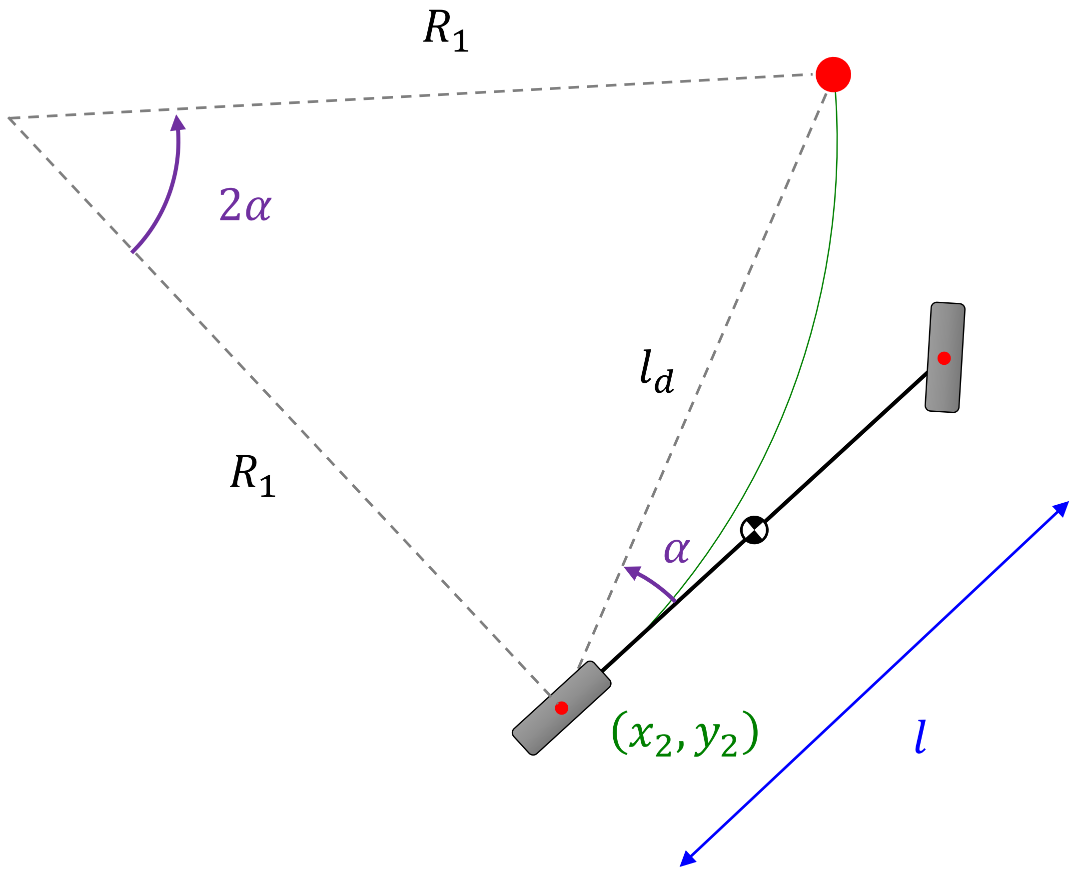
Kinematic Steering Technique

Overview



- The **pure pursuit algorithm** connects the rear axle of the vehicle with a given target waypoint by means of a lookahead distance l_d.
- The goal is to calculate the trajectory curvature among these points.
- The steering angle δ is obtained from:
  - Target point location
  - Angle α between the vehicle heading and the lookahead direction

Approach



Sines law

$$\frac{l_d}{\sin(2\alpha)} = \frac{R_1}{\sin\left(\frac{\pi}{2} - \alpha\right)}$$

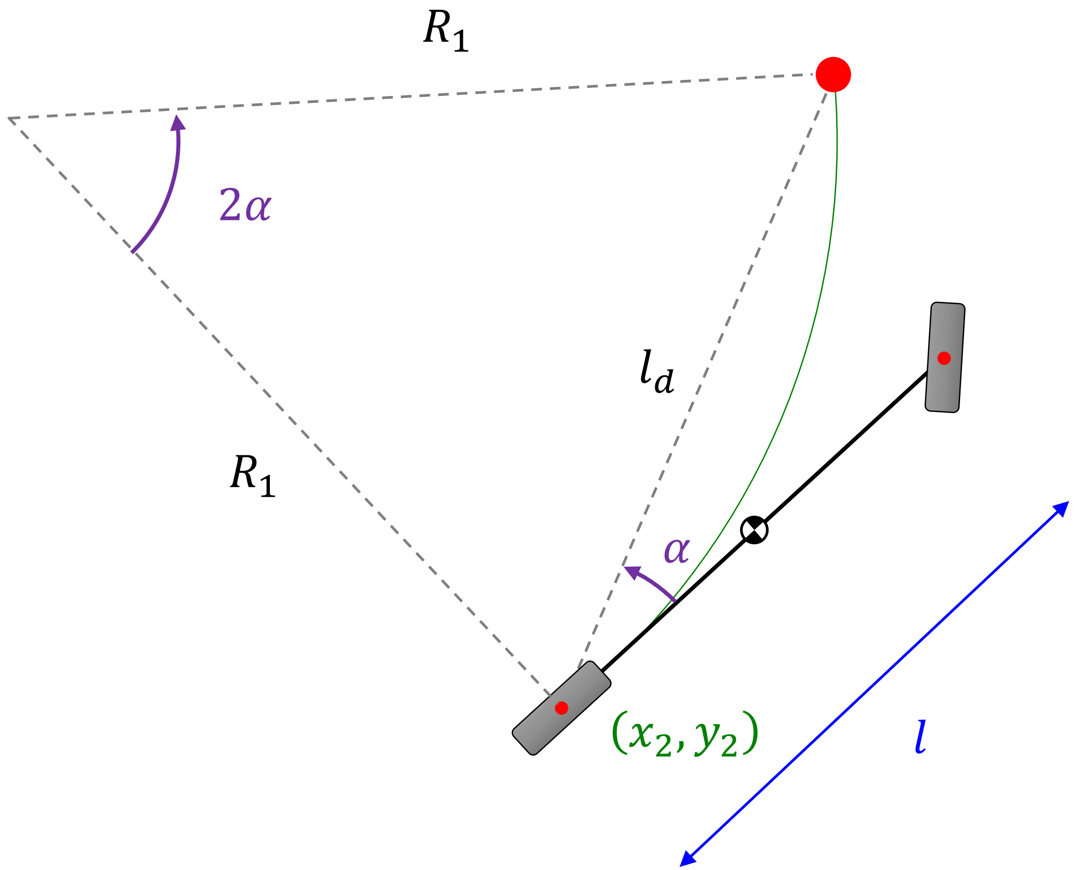
$$\frac{l_d}{2\sin\alpha\cos\alpha} = \frac{R_1}{\cos\alpha}$$

$$\frac{l_d}{2\sin\alpha} = R_1$$

Path curvature

$$\kappa = \frac{1}{R_1} = \frac{2\sin\alpha}{l_d}$$

Approach



Path curvature

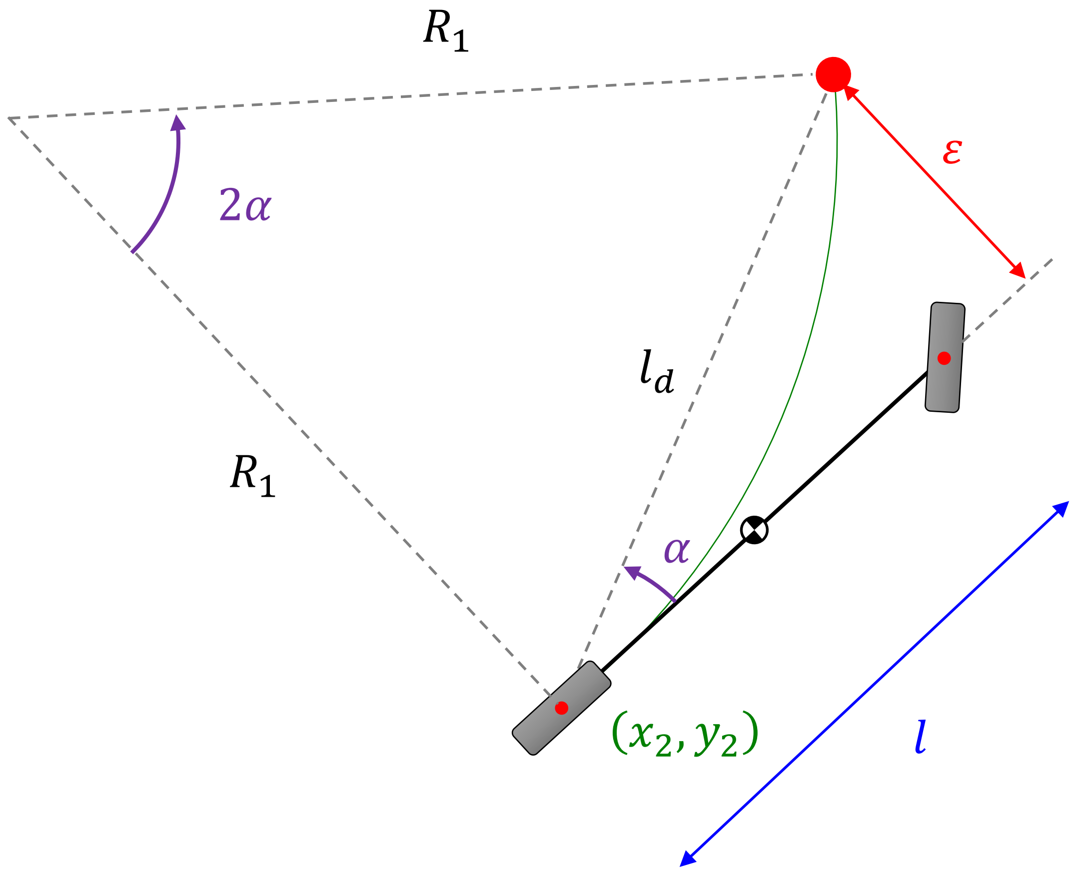
$$\kappa = \frac{1}{R_1} = \frac{2\sin\alpha}{l_d}$$

Bicycle model (rear axle)

$$\tan\delta = \frac{l}{R_1} \rightarrow \delta = \arctan(\kappa l)$$

$$\delta = \arctan\left(\frac{2l\sin\alpha}{l_d}\right)$$

Cross-Track Error



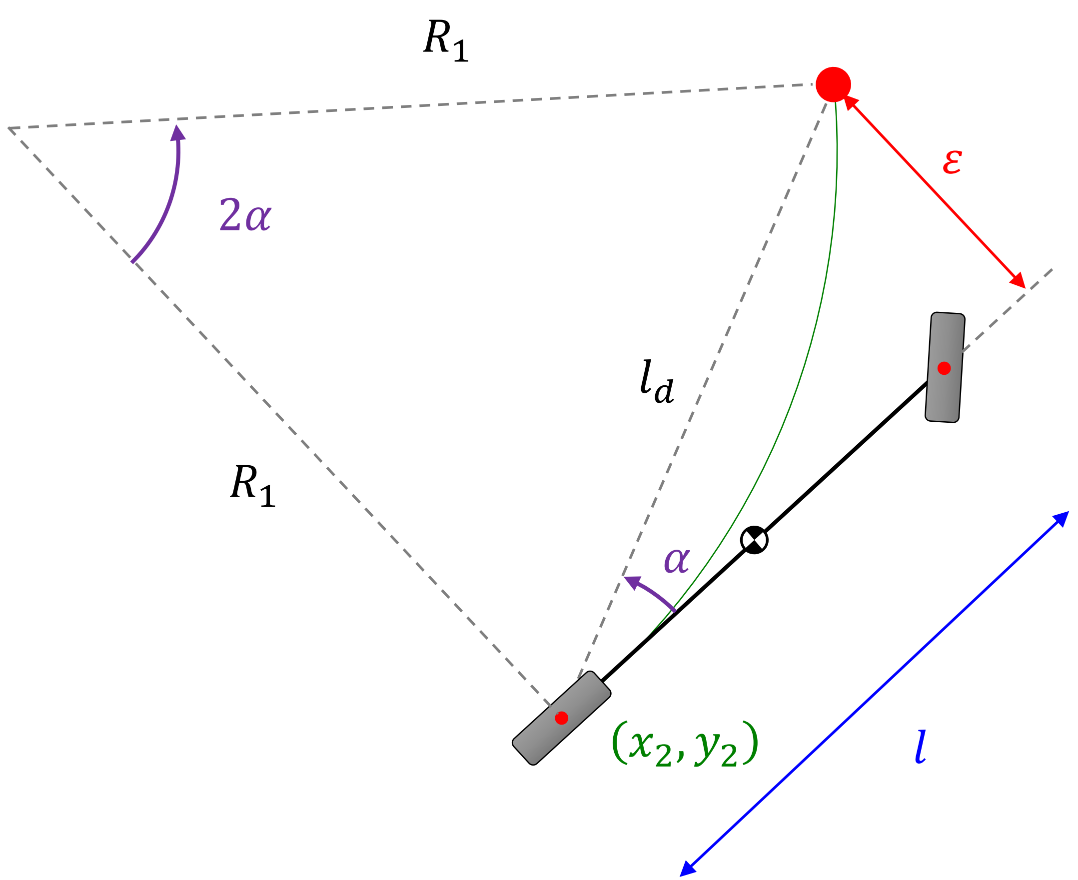
- The cross-track error ε is defined as the lateral distance between the vehicle heading and the target waypoint.

$$\sin\alpha = \frac{\varepsilon}{l_d}$$

$$\kappa = \frac{2\sin\alpha}{l_d} = \frac{2}{l_d^2}\varepsilon$$

- Remember that $\delta = f(\kappa)$. Hence, the steering input depends on the cross-track error and a proportionality constant $2/l_d^2$.

Setting the Lookahead Distance



- The lookahead distance l_d plays a fundamental role in selecting a proper steering input δ.
- The faster the vehicle travels, the further this distance must be set to yield effective commands to follow the target path.
- **Idea:** Set l_d proportional to the vehicle longitudinal speed v.

$$l_d = K_d v$$

$$\delta = \arctan\left(\frac{2l \sin\alpha}{K_d v}\right)$$

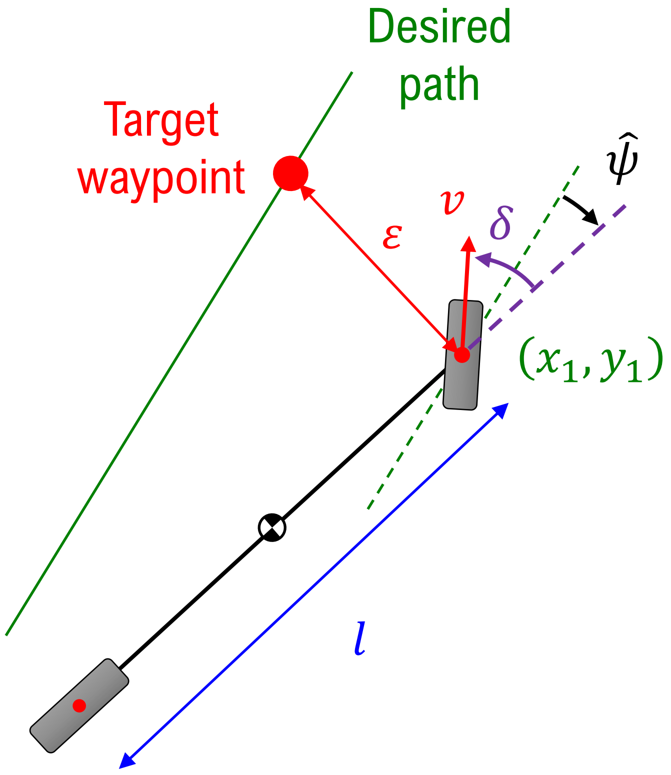
Stanley Control

Kinematic Steering Technique

Overview

- The **Stanley algorithm** defines a steering law to
  - Correct the heading error
  - Correct the position error
  - Constrain δ to feasible bounds
- The control approach is referred to the front axle.
- The Stanley algorithm was developed by Stanford University for the 2005 DARPA Grand Challenge.

Heading Control Requirements



Alignment contribution

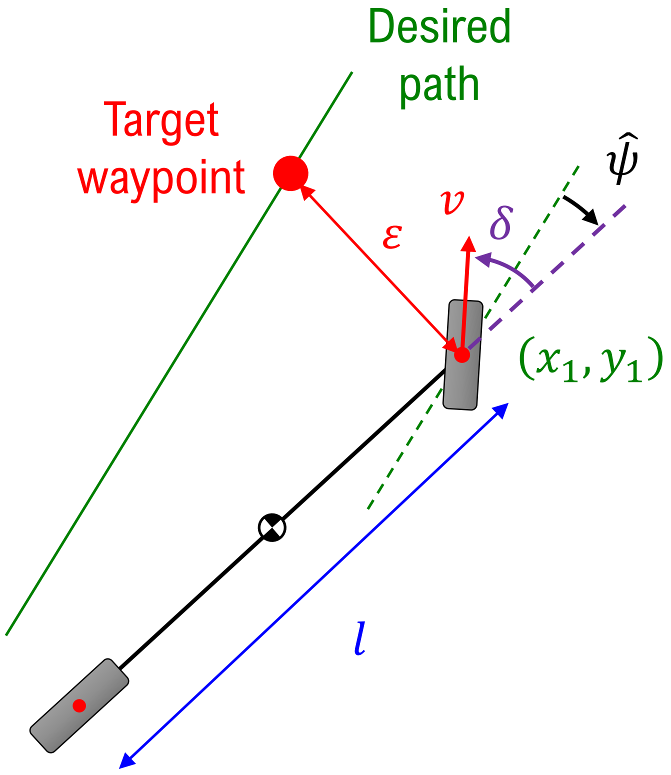
- Apply a steering input to align heading with desired heading.

$$\delta(t) = \hat{\psi}(t)$$

- *Remark 1: $\hat{\psi}$ is the yaw angle with respect to the desired path.*

- *Remark 2: Notice that the sense of δ and $\hat{\psi}$ are opposite.*

Heading Control Requirements



Cross-track error elimination

- Apply a steering input

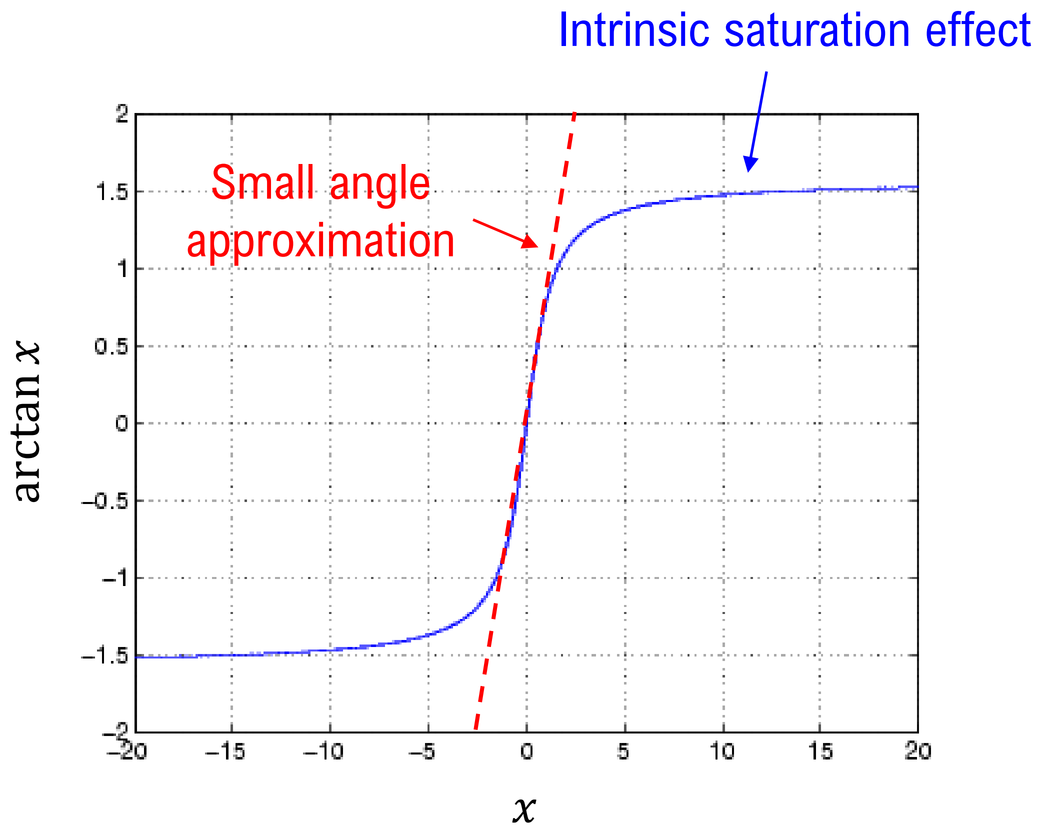
$$\delta(t) = \arctan\left(\frac{K\varepsilon(t)}{v(t)}\right)$$

For small arguments: $\delta(t) \approx \frac{K\varepsilon(t)}{v(t)}$

  - Directly proportional to the cross-track error ε
  - Inversely proportional to longitudinal speed v
  - Gain K is tuned heuristically

Command bounds $\delta(t) \in [\delta_{min}, \delta_{max}]$

Cross-Track Elimination



Stanley

$$\delta(t) = \arctan\left(\frac{K\varepsilon(t)}{v(t)}\right)$$

Pure pursuit

$$\delta = \arctan\left(\frac{2l\varepsilon(t)}{K_d^2 v(t)^2}\right)$$

Steering Law

Command combination

$$\delta(t) = \hat{\psi}(t) + \arctan\left(\frac{K\varepsilon(t)}{K_s + v(t)}\right)$$

$$\delta(t) \in [\delta_{min}, \delta_{max}]$$

- The cross-track error control law can be modified with:
  - Softening constant K_s to improve numerical stability when $v \approx 0$
  - Control law enhancement by adding integral and derivative terms

Example

Stanley Control in Python

Parameters

Stanley Control Tuning

- Proportional constant $K = 2.5$

Vehicle / Inputs

- Wheelbase $l = 1 \text{ m}$
- Maximum steering angle of 25 deg in both directions.
- Constant longitudinal speed v. Try 2, 5 and 10 m/s.

Trajectory

- Move along x axis only. At 20 meters, introduce a step variation towards 5 meters in the y axis
- Initial yaw angle $\psi = 20 \text{ deg}$
- Initial cross track error $\varepsilon = 0$

Euler discretization

$$\frac{x_k - x_{k-1}}{\Delta t} \cong \dot{x} = f(x, u)$$

$$x_k = x_{k-1} + \Delta t f(x, u)$$

*Implement front-axle kinematics