A Streamlined Nonlinear Path Following Kinematic Controller

Jesús Manuel de la Cruz García
(jmcruz@ucm.es)

Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

UNED, 24 de marzo de 2015

Outline

- Motion Control Problems of Autonomous Vehicles
- Review of Path-Following Algorithms
- A Streamlined Nonlinear Path Following Kinematic Controller
- Simulations

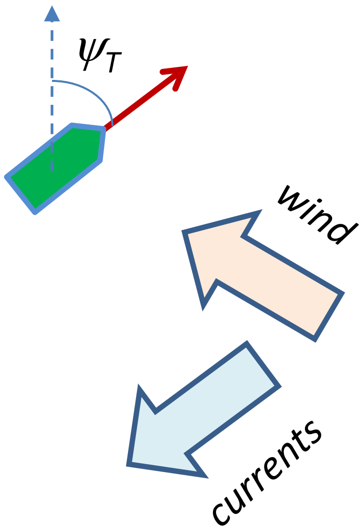
SISTEMA AUTONOMO PARA LA LOCALIZACION Y ACTUACION ANTE CONTAMINANTES EN EL MAR

DPI2013-46665-C2-1/2-R

Motion control problems of autonomous vehicles

- Point stabilization

Design of control laws that stabilize the vehicle at a given target point with a desired orientation.

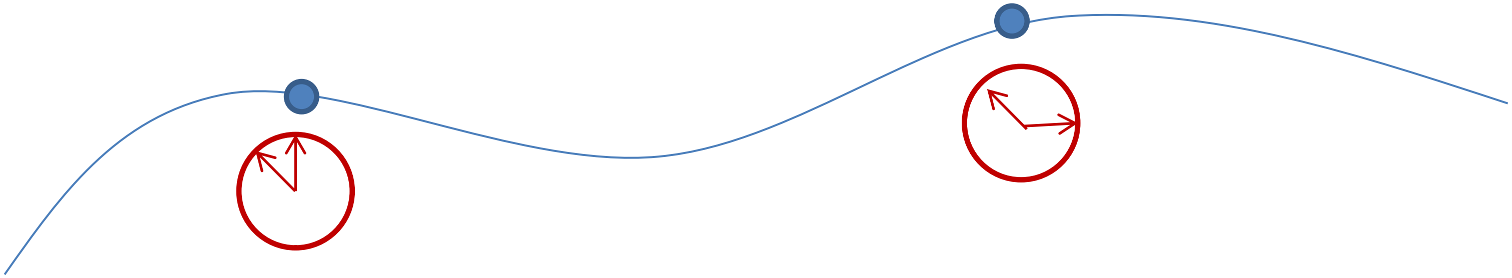


Motion control problems of autonomous vehicles

- Trajectory tracking

Design of control laws that force a vehicle to reach and follow a geometric path with an associated timing law.

Usually, tracking problems for autonomous vehicles are solved by designing control laws that make the vehicles track pre-specified feasible “state-space” trajectories, i.e., trajectories that specify the time evolution of the position, orientation, as well as the linear and angular velocities, all consistent with the vehicles’ dynamics

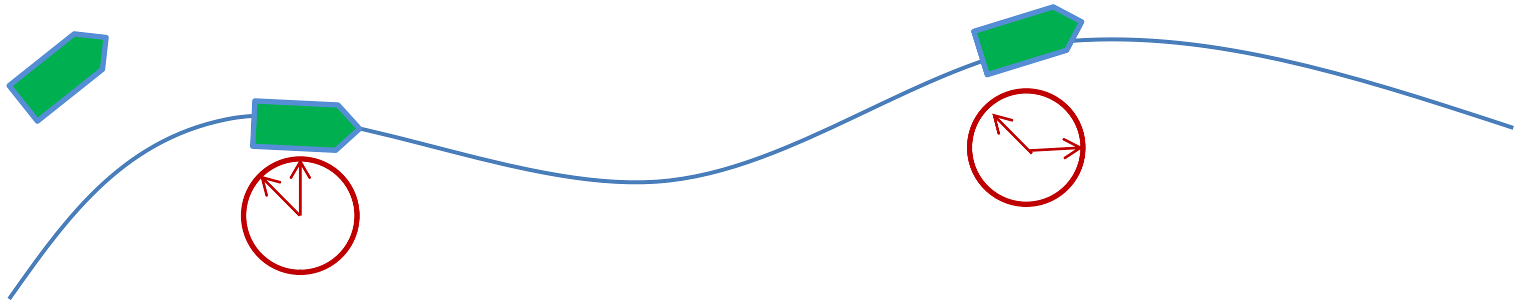


Motion control problems of autonomous vehicles

- Trajectory tracking

  Design of control laws that force a vehicle to reach and follow a geometric path with an associated timing law.

  Usually, tracking problems for autonomous vehicles are solved by designing control laws that make the vehicles track pre-specified feasible “state-space” trajectories, i.e., trajectories that specify the time evolution of the position, orientation, as well as the linear and angular velocities, all consistent with the vehicles’ dynamics



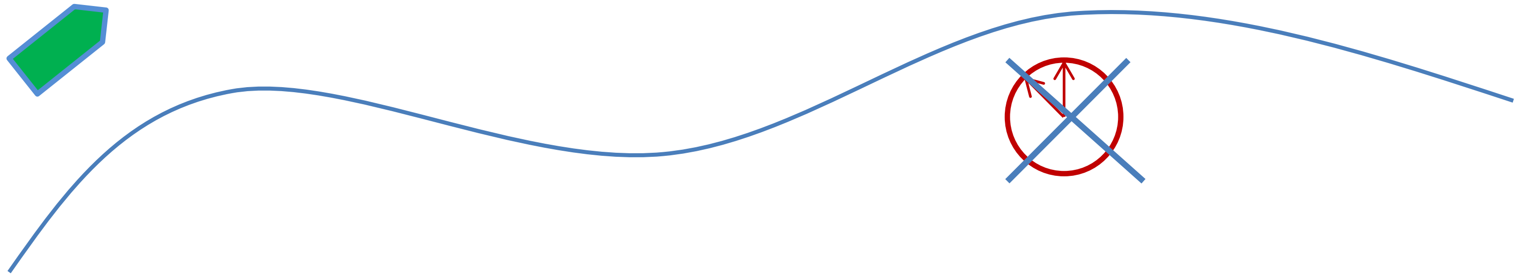
Motion control problems of autonomous vehicles

- Path following

Design of control laws that force a vehicle to converge to and follow a path that is specified without a temporal law.

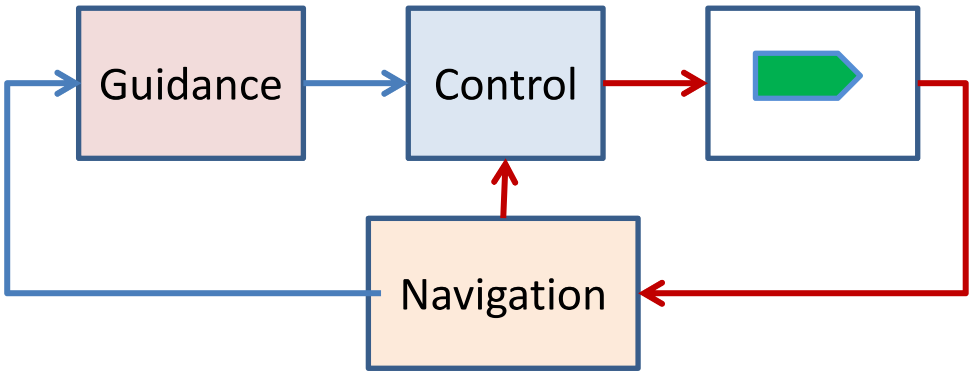
This problem can be expressed by the following two task objectives:

- ***Geometric Task :*** make the position of the vehicle converge to and follow a desired geometrical path.
- ***Dynamic Task:*** make the vehicle satisfy a dynamic assignment along the path, e.g. the speed of the vehicle converge to and track a desired speed assignment (*maneuvering*)



Path Following Algorithms

- The path following loop is divided in an inner control loop and an outer guidance loop.



› The inner loop controller stabilizes the vehicle dynamics

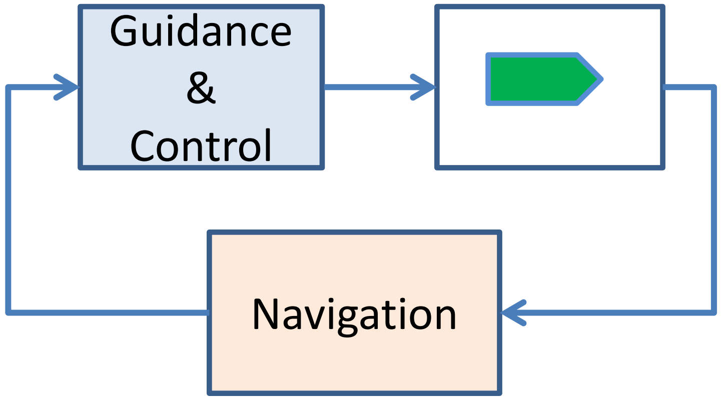
› The outer loop controls the vehicle kinematics and computes reference commands to the inner loop controller, providing path-following capabilities.

› If there is adequate frequency separation between the guidance and control systems the combined scheme will perform as specified separatelly.

›This structure is the usual one when the vehicle comes equipped with an autopilot.

Path Following Algorithms

- Integrated guidance and control are designed simultaneously



A survey of control and guidance and control algorithms for vessels can be found in:

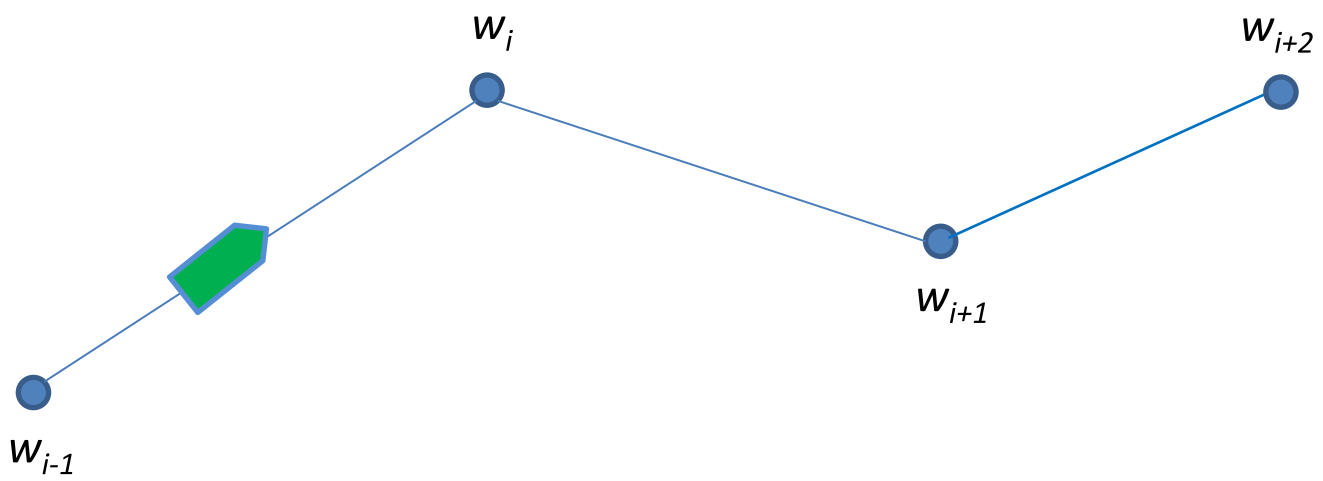
Automática marina: una revisión desde el punto de vista del control
J. M. de la Cruz García , J. Aranda Almansa, J.M. Girón Sierra,
Revista Iberoamericana de Automática e Informática industrial RIAI,
vol. 9, pp. 205-218, 2012.

Paths

- A waypoint path is an ordered sequence of waypoints:

$$W = \{w_1, w_2, \ldots, w_N\},$$

$$w_i = (w_{n,i}, w_{e,i}, w_{d,i})^T \in R^3 \quad \text{or} \quad w_i = (w_{n,i}, w_{e,i})^T \in R^2$$



Dubins Paths

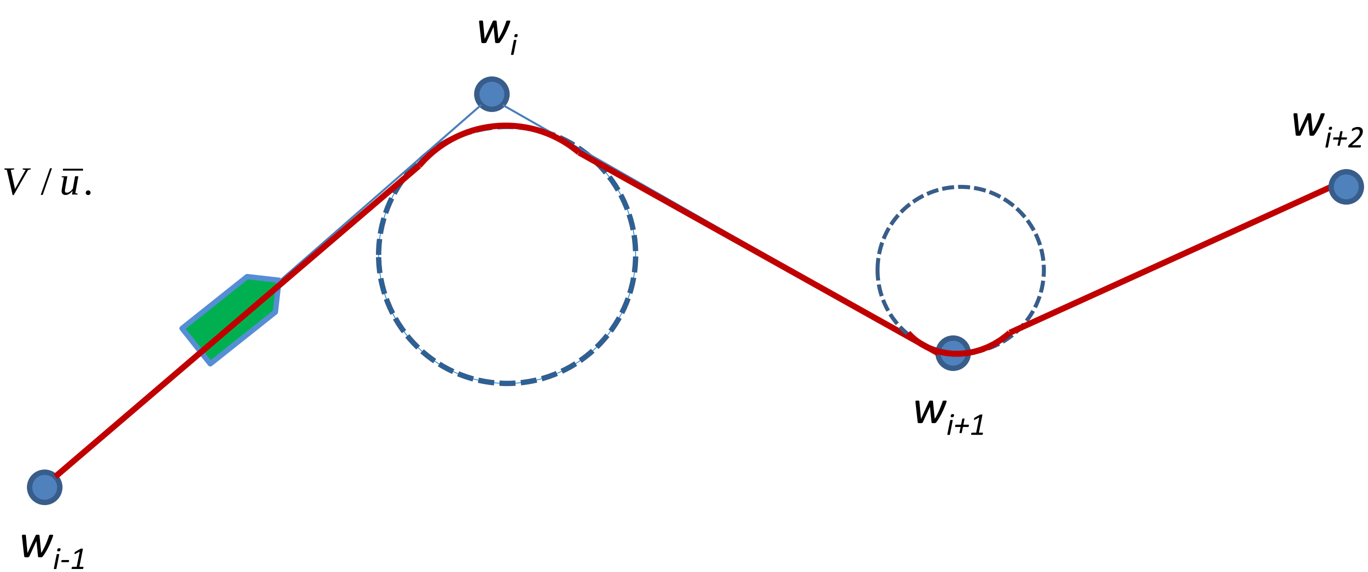
- For a vehicle with kinematics

$$\dot{x} = V\cos\psi$$
$$\dot{y} = V\sin\psi$$
$$\dot{\psi} = u, \quad u \in [-\bar{u}, \bar{u}].$$

(x, y) position
ψ heading
V speed

moving at constant speed V the time-optimal path (shortest path) between two different configurations is a path formed by straight-lines and circular arc segments.

$$R_{\min} = V / \bar{u}.$$



Parametrized Path

- A parametrized path is a geometric curve $p_d(\theta)$ parametrized by a continuous path variable θ.

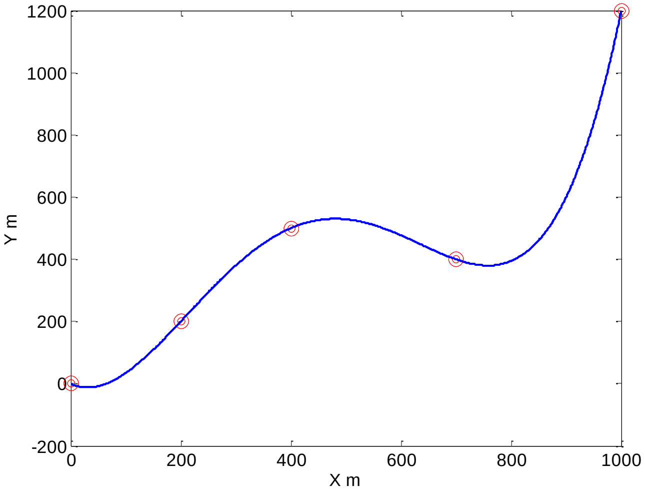
$$p_d(s) = [x_d(\theta), y_d(\theta), z_d(\theta)]^T \quad \text{or} \quad p_d(\theta) = [x_d(\theta), y_d(\theta)]^T$$

- Given a set of waypoints $W = \{w_1, w_2, \ldots, w_N\}$ a parametrized path can be generated using spline or polynomial interpolation methods.

  – Example: Cubic polynomial for $p_d(\theta) \in R^2$

$$x_d(\theta) = a_0 + a_1\theta + a_2\theta^2 + a_3\theta^3$$
$$y_d(\theta) = b_0 + b_1\theta + b_2\theta^2 + b_3\theta^3$$

Time independent path.



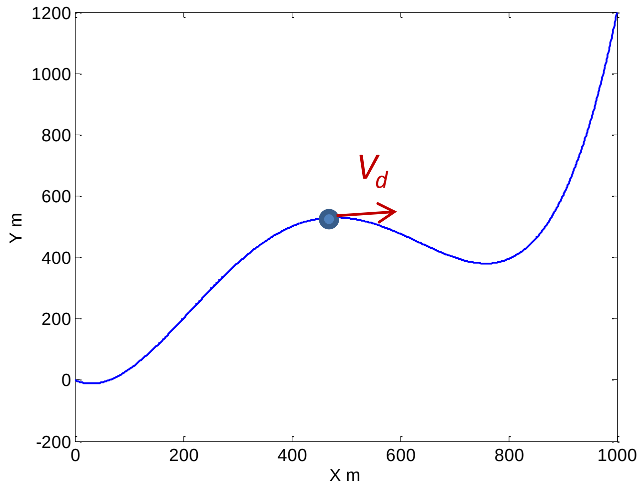
Parametrized Path: Reference Trajectory

- The time independent path can be transformed to a time varying trajectory by defining *a speed profile along the path*, $V_d(t)$,

$$x_d'(\theta) \triangleq \frac{dx_d(\theta)}{d\theta} \quad \rightarrow \quad \dot{x}_d(t) = x_d'(\theta)\dot{\theta}(t)$$

$$y_d'(\theta) \triangleq \frac{dy_d(\theta)}{d\theta} \quad \rightarrow \quad \dot{y}_d(t) = y_d'(\theta)\dot{\theta}(t)$$

$$V_d(t) = \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad \rightarrow \quad \dot{\theta}(t) = \frac{V_d(t)}{\sqrt{x_d'(\theta)^2 + \dot{y}_d'(\theta)^2}}$$

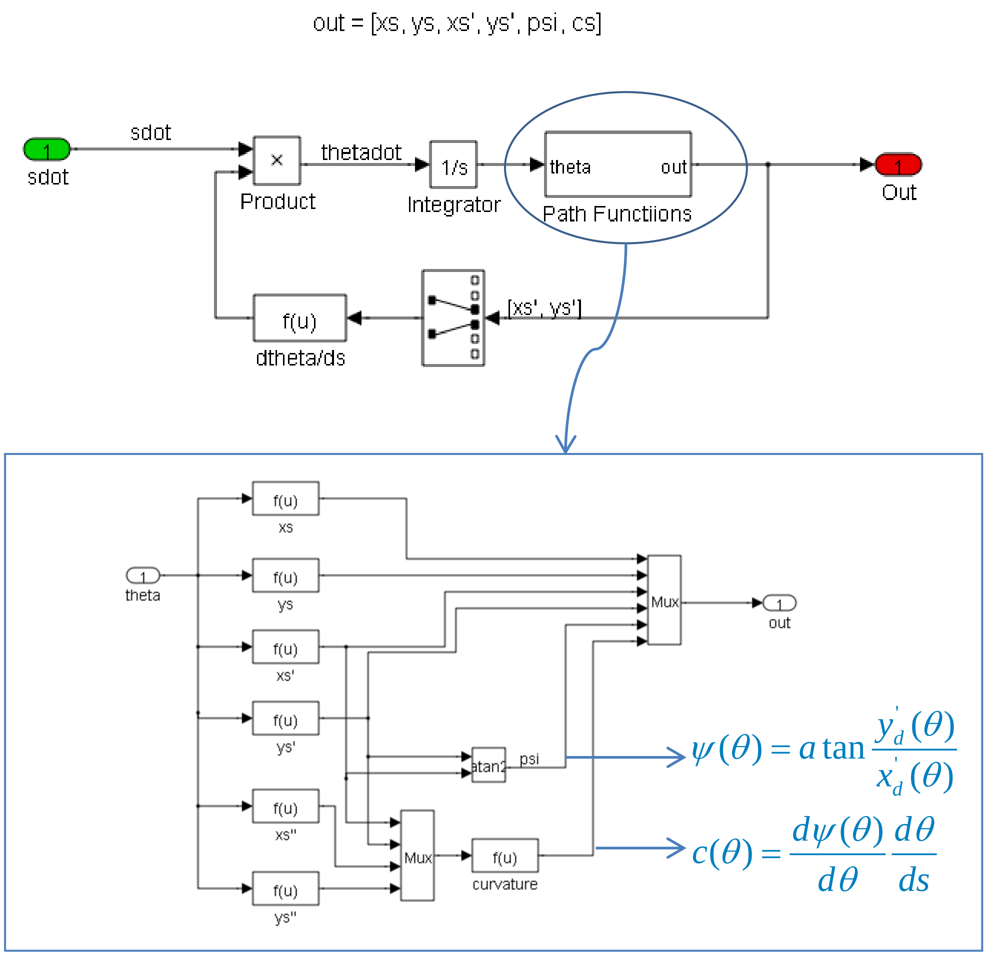


Let s be the length of the path, then

$$ds = \sqrt{x_d'(\theta)^2 + \dot{y}_d'(\theta)^2}\, d\theta \quad \rightarrow \quad \frac{d\theta}{ds} = \frac{1}{\sqrt{x_d'(\theta)^2 + \dot{y}_d'(\theta)^2}}$$

$$s(t) \triangleq \int_0^t V(\tau)\, d\tau = \int_0^t \sqrt{x_d'(\theta)^2 + \dot{y}_d'(\theta)^2}\, \dot{\theta}(\tau)\, d\tau$$

Simulink model for path generation



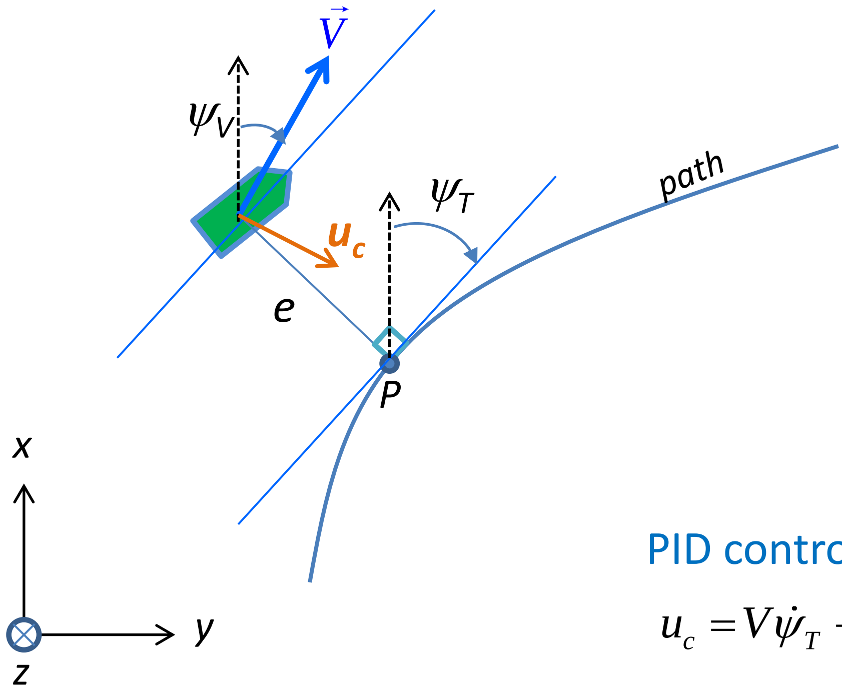
$$\psi(\theta) = a\tan\frac{y_d^{'}(\theta)}{x_d^{'}(\theta)}$$

$$c(\theta) = \frac{d\psi(\theta)}{d\theta}\frac{d\theta}{ds}$$

Outer guidance controllers for path following

- PID controllers
- Virtual Point Tracking
- Line of Sight Guidance
- A Streamlined Nonlinear Path Following Kinematic Controller

PID Controller



Suppose : V is constant

$\psi_T, \dot{\psi}_T$ are given

The control objectives are

$$e \to 0$$

$$\psi_V \to \psi_T$$

Command signal:

$$u_c = V\dot{\psi}_V \quad \to \quad \dot{\psi}_V = u_c / V$$

PID controller

$$u_c = V\dot{\psi}_T + K_D\,\dot{e} + K_P\,e + K_I \int_0^t e\,dt$$

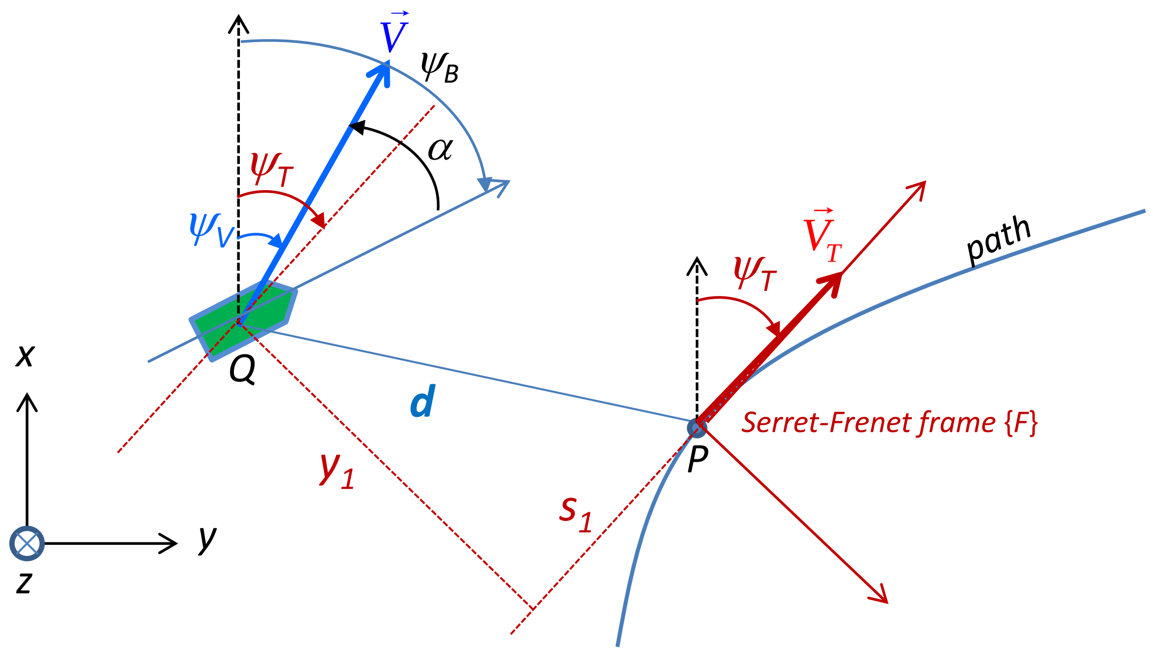
$$\text{if} \quad \psi_V \approx \psi_T \to \dot{e} \approx V(\psi_V - \psi_T)$$

Problems

-Determine gains K_P , K_D , K_I (LQR, root locus, …)

-Determine point P (might be indeterminate)

Virtual Point Tracking: Serret-Frenet Frame



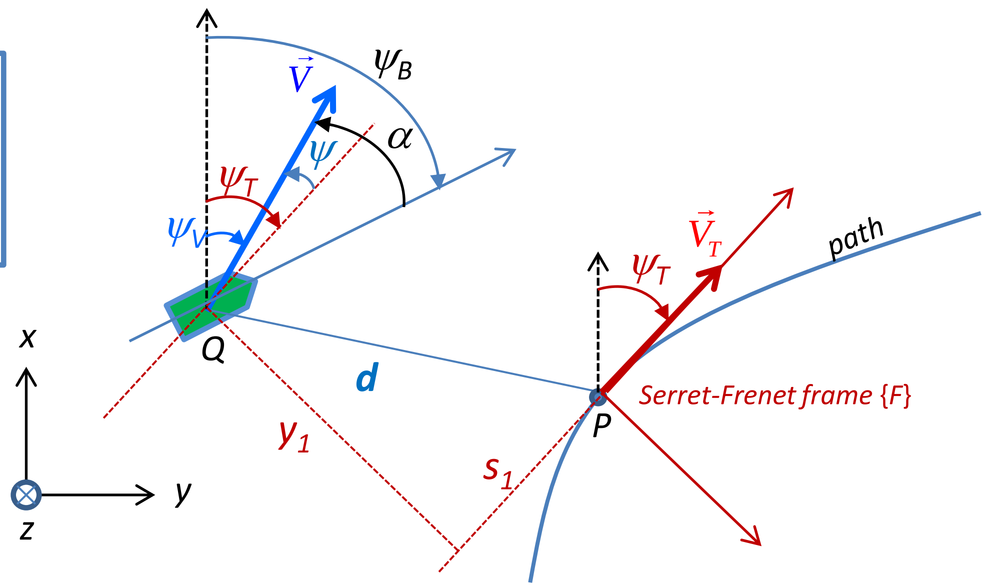
- Point P defines a point on the path where a Serret-Frenet frame $\{F\}$ is defined. $\{F\}$ plays the role of a virtual point or target that should be tracked by the vehicle Q.

- P is not the point on the path closest to Q but a point that is made evolved according to a conveniently defined control law.

Virtual Point Tracking: Control Signals

Control signals: $\dot{\psi}_B \; (V\dot{\psi}_B), \;\; \dot{s}$

$$\dot{\psi}_T = \dot{s}\,\kappa(s)$$
$$\psi = \psi_V - \psi_T$$
$$\psi_V = \psi_B + \alpha$$



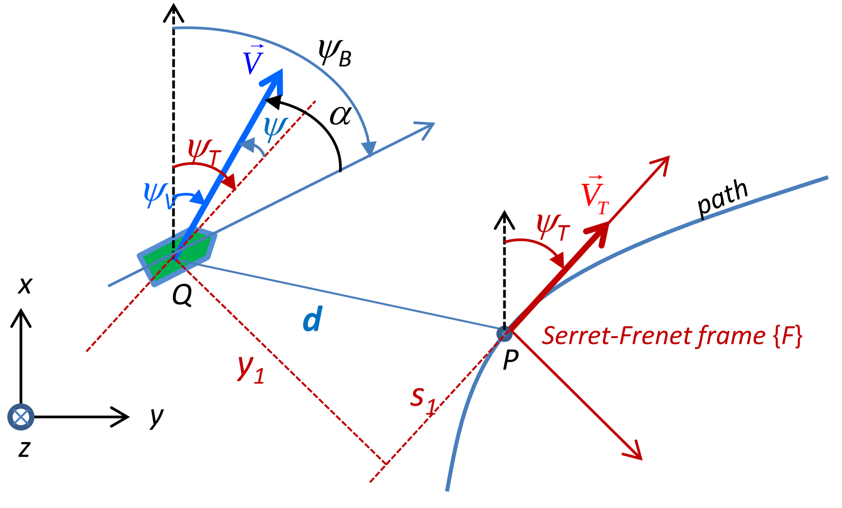
- s_1 along-track error, y_1 cross-track error, ψ course error
- s lenght that the virtual point has moved along the path
- $\kappa(s)$ path curvature
- the path is parametrized by s

Virtual Point Tracking: Kinematic Model

Point $Q = [s_1, y_1]^T$ in $\{F\}$ evolves according to the equations

$$\dot{s}_1 = -\dot{s}(1 - y_1\,\kappa(s)) + V\cos\psi$$
$$\dot{y}_1 = -\dot{s}\,s_1\,\kappa(s) + V\sin\psi$$
$$\dot{\psi} = \dot{\psi}_V - \dot{\psi}_T = \dot{\psi}_B + \dot{\alpha} - \dot{s}\,\kappa(s)$$

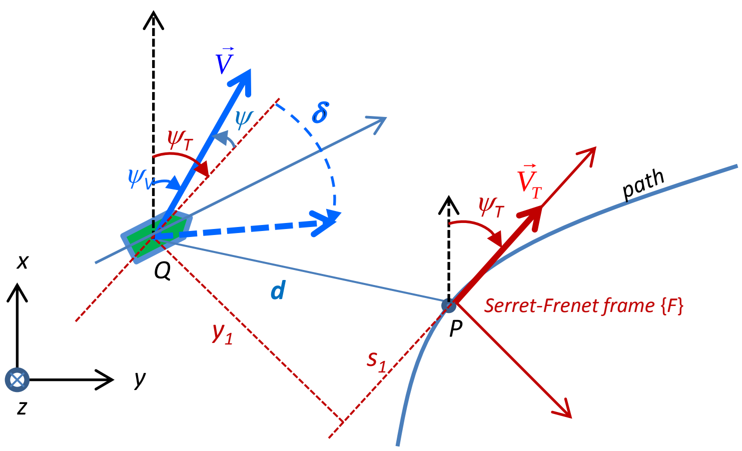
The objective is to drive the error coordinate (s_1, y_1, ψ) to zero with controls: $\dot{\psi}_B,\ \dot{s}$



Equilibrium point $(s_1, y_1, \psi) = (0,0,0)$

$\dot{s} = V$

Virtual Point Tracking: Kinematic Path Following Controller



$$\dot{\psi} = \dot{\delta} - K_1(\psi - \delta) \quad (1)$$
$$\dot{s} = V\cos\psi + K_2 s_1 \quad (2)$$

(1) $\dot{\psi}_B = -\alpha + \dot{s}\kappa(s) + \dot{\delta} - K_1(\psi - \delta)$

Try to bring y_1 and ψ to 0.

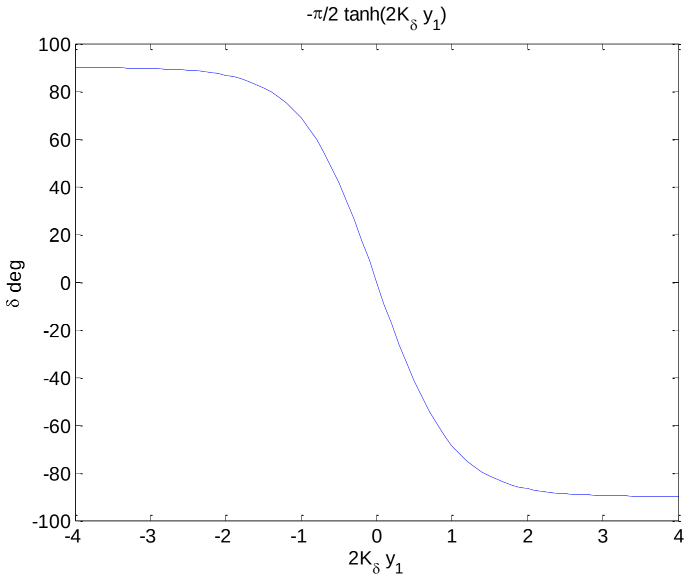
(2) Try to bring s_1 to 0.

- δ is a desired approach angle
- K_1, K_2 are design parameters
- ψ_B and ψ_V can be obtained from an IMU

Virtual Point Tracking: Approach Angle

δ can be any function of y_1 satisfying $y_1 \delta(y_1) \le 0$

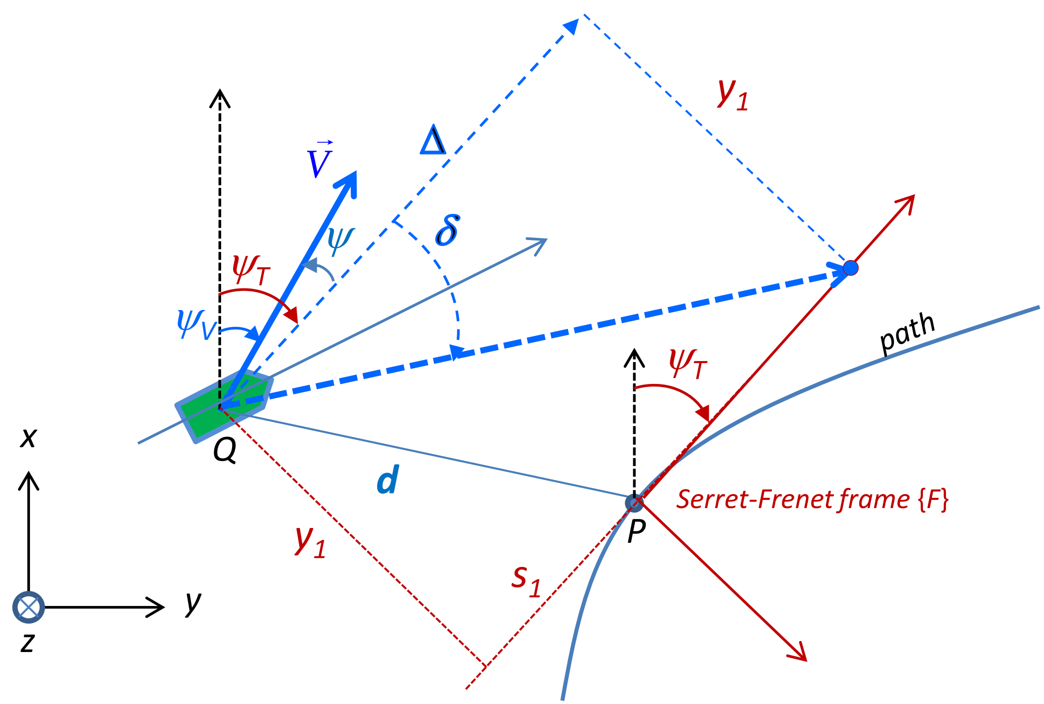
$$\delta = -\psi_\delta \tanh(2K_\delta y_1) = -\dot{\psi}_d \frac{e^{2K_\delta y_1} - 1}{e^{2K_\delta y_1} + 1}, \quad 0 < \psi_\delta < \pi / 2, \; 0 < K_\delta$$



- Two design parameters: K_δ, ψ_δ

- The equilibrium point (s_1, y_1, ψ)=(0, 0, 0) is *Uniformly Global Asymptotically Stable (UGAS)* and *Uniformly Local Exponentially Stable (ULES)*

Line of Sight Guidance: Marine Vehicles*



- The vehicle velocity vector is directed *toward a point ahead of the direct projection of the craft to the tangent*, located at a distance $\Delta > 0$.

"Practice of good helmsman when steering a boat"

- The approach angle is now

$$\delta = \arctan(-y_1 / \Delta)$$

- Since $y_1 \delta \leq 0$ for all y_1 the previous stability properties are kept.

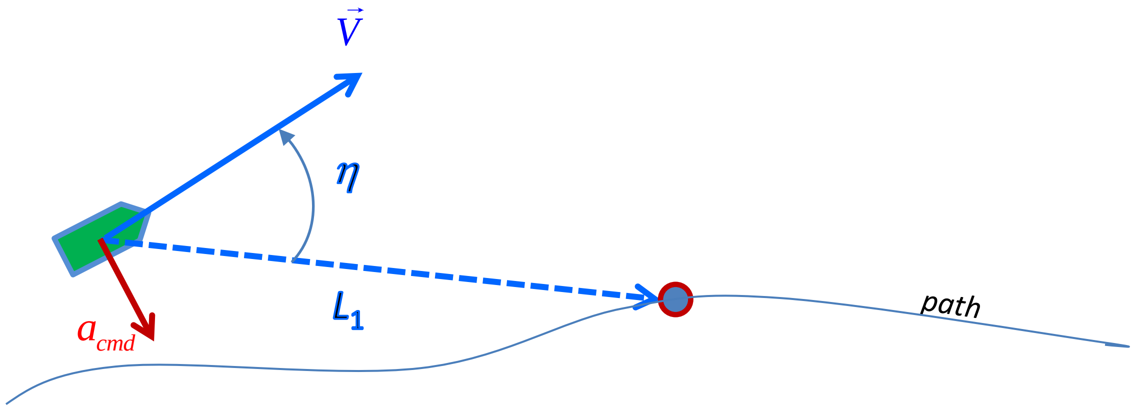
- Three design parameters K_1, K_2 , Δ.

- No need to compute the curvature of the path

* Papoulias (1992), Breivik and Fossen (2004), Børhaugh and Pettersen (2006)

Line of Sight Guidance: Air Vehicles

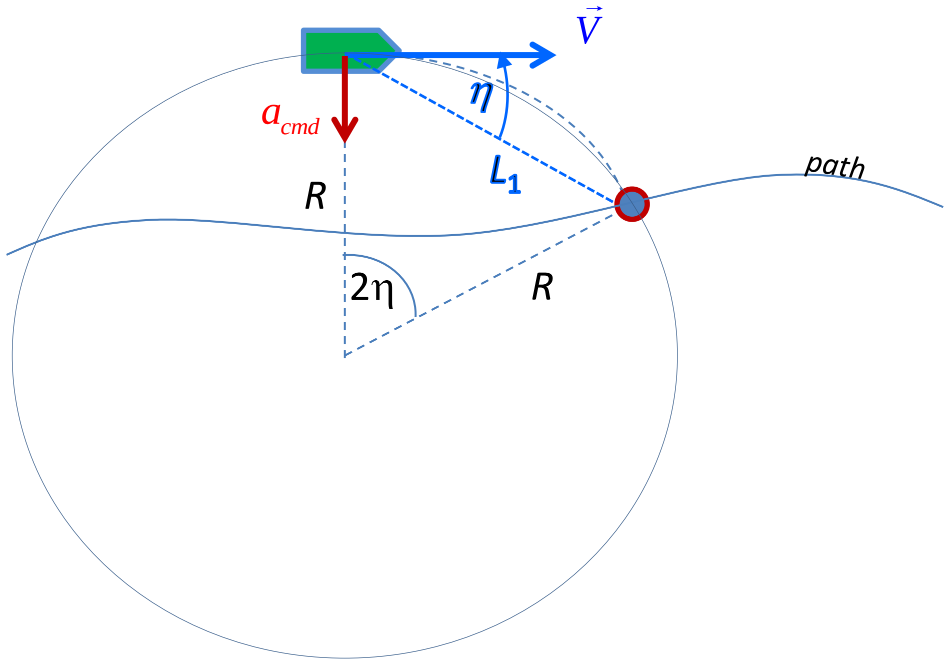
- Originally developed for missile guidance
- Introduced by Amidi (1991) for WMR and Adopted for UAVs in Park et al. (2004,2007)
- A reference point *P* on the desired path at a *constant distance* L_1 is designated
- A lateral acceleration command is generated according to the direction of *P* relative to vehicle's velocity



Line of Sight Guidance: L_1 Guidance Law

- The acceleration command is equal to the centripetal acceleration required to follow a circular path that passes through the reference point and is tangent to the vehicle velocity vector

$$L_1 = 2R\sin(\eta) \quad \rightarrow \quad a_{cmd} = \frac{V^2}{R} = \frac{2V^2}{L_1}\sin(\eta) = V\omega_V$$



Line of Sight Guidance: L_1 Guidance Law Properties

- The law uses instantaneous ground speed and compensates naturally for wind
- It has an element of anticipation of the desired path, enabling tight tracking of curved trajectories
- Only one parameter L_1 to tune.
- Lyapunov stability is proven for tracking circular paths when $L_1 < R$, and straight lines
- It approximates a PD controller when following straight-line paths
- For small perturbations when following a path, the cross track error and course error dynamics behave as a second order system

$$\ddot{y}_1 + \frac{2V}{L_1}\dot{y}_1 + \frac{2V^2}{L_1^2}y_1 = \frac{2V^2}{L_1^2}y_{ref}, \quad \psi \approx \dot{y}_1 / V$$

$$\zeta = 0.707$$

$$\omega_n = \frac{\sqrt{2}V}{L_1}$$

$$\tau_L = \frac{1}{\zeta\omega_n} = \frac{L_1}{V}$$

- *The L_1 intercept can be undefined*

Line of Sight Guidance: L_1 Guidance Law Properties

- If the control law and the natural vehicle dynamics are sufficiently faster than the guidance law, no appreciable dynamic interactions between the two schemes should be expected†.
- If this is not the case stability of the combined guidance and control law is no longer guaranted†.
- If the dynamic of the inner control law can be characterized by a time contant τ_{il}, it can be seen that the guidance system is marginally stable when $\tau_L = \tau_{il}$, so it is important to ensure ‡

$$\tau_L > \tau_{il}$$

A value of $\tau_L \approx 3\ \tau_{il}$ or $4\tau_{il}$ should be chosen to ensure satisfactory transient response.

- L_1 can be adapted to the ground speed to keep a constant τ_L^*

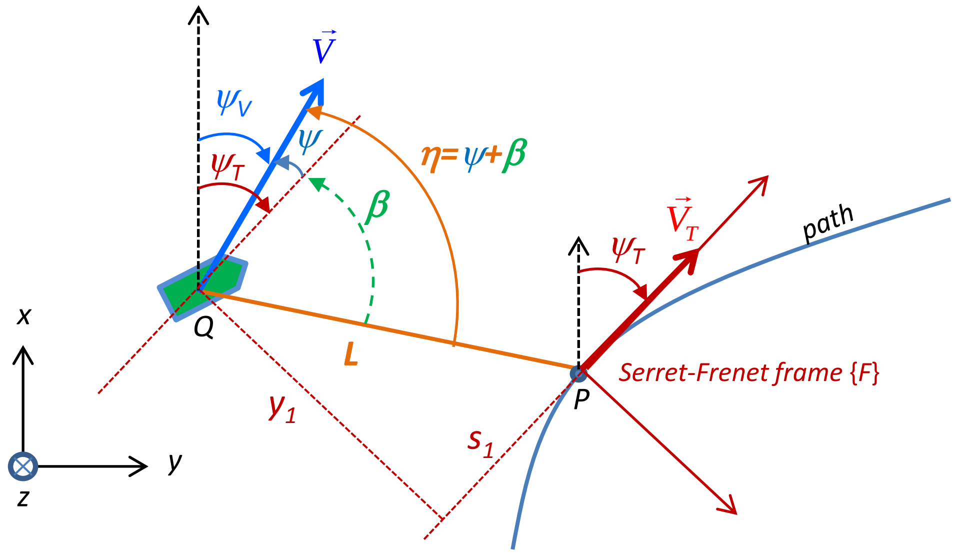
$$L_2 = \tau_L^* V_g$$

with guidance law

$$a_{cmd} = \frac{2V_g}{\tau_L^*}\sin(\eta)$$

†Papoulias, 1992, ‡ Curry et al. 2013.

A Streamlined Nonlinear Guidance Law*



$$\dot{s}_1 = -\dot{s}(1 - y_1\kappa(s)) + V\cos\psi$$

$$\dot{y}_1 = -\dot{s}\,s_1\,\kappa(s) + V\sin\psi$$

$$\dot{\psi} = \dot{\psi}_V - \dot{\psi}_T = \dot{\psi}_V - \dot{s}\,\kappa(s)$$

Guidance Law

$$\dot{\psi}_V = \begin{cases} -\dfrac{2V}{L}\sin(\eta), & |\eta| \le \dfrac{\pi}{2} \quad \text{(2a)} \\ -\dfrac{2V}{L}\mathrm{sign}(\eta), & |\eta| > \dfrac{\pi}{2} \quad \text{(2b)} \end{cases}$$

$$\dot{s} = V\cos\psi + K(s_1 + L) \quad \text{(3)}$$

- (2a-2b) tryes to bring the cross-track error and the course error to zero
- (3), $K > 0$, tryes to make the vehicle follow the moving reference point with a constant along-track error *L*.
- We do not consider a reference point on the path at a distance *L* from the vehicle, but *a desired distance from the vehicle to the reference point on the path*

*J.M. de la Cruz, J.A López-Orozco, E. Besada-Portas, J-Aranda ICRA 2015.

Analysis of the Circular and Straight-Line Path Following

- If we consider a circular path of radius $R = \kappa(s)^{-1}$, the stationary conditions yield the relation

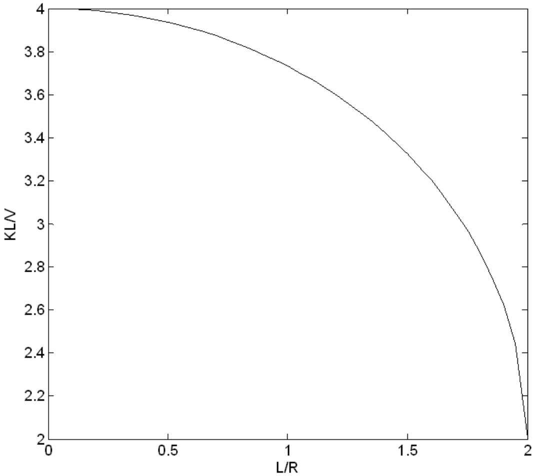
$$\frac{KL}{V} = \frac{1-\cos 2\beta^*}{1-\cos \beta^*} \qquad (4)$$

$$\sin \beta^* = \frac{L}{2R}$$

- The dimentionless quantity *KL/V* is a function of the relation *L/R,* therefore the stationary point depends only on *L* and *R* and not on *V.*

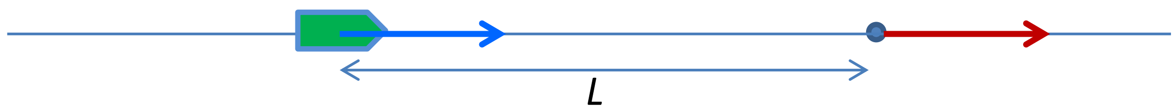
- (4) gives a constraint that determines *K* adaptively as a function of the present curvature of the path, ground speed and the chosen *L.*

- If the time constant $\tau_L = L/V$ is specified then $K \in [2,\ 4]*1/\tau_L$.



Straight-Line Path Following

- Stationary point $s_1^* = -L, \quad y_1^* = 0, \quad \psi^* = 0, \quad \eta^* = 0, \quad \dot{s}^* = V.$



- The equilibrium point is *Uniformly Global Asymptotically Stable* and *Uniformly Local Exponentially Stable* (Lyapunov).

- Dynamics of the along-track error $\dot{s}_1 = -K(s_1 + L).$

- Linearazing the equations of the cross-track error and course error about de e.p. a second order time is obtained with

$$\zeta = 1/\sqrt{2}, \; \omega_n = \frac{\sqrt{2}V}{L}$$

and , since *R=∞*

$$\frac{KL}{V} = 4 \quad \Rightarrow \quad K = 4\frac{V}{L} = 2\sqrt{2}\,\omega_n$$

Circular Path Following: Stationary Points

$$s_1^* = R\sin\psi^*$$

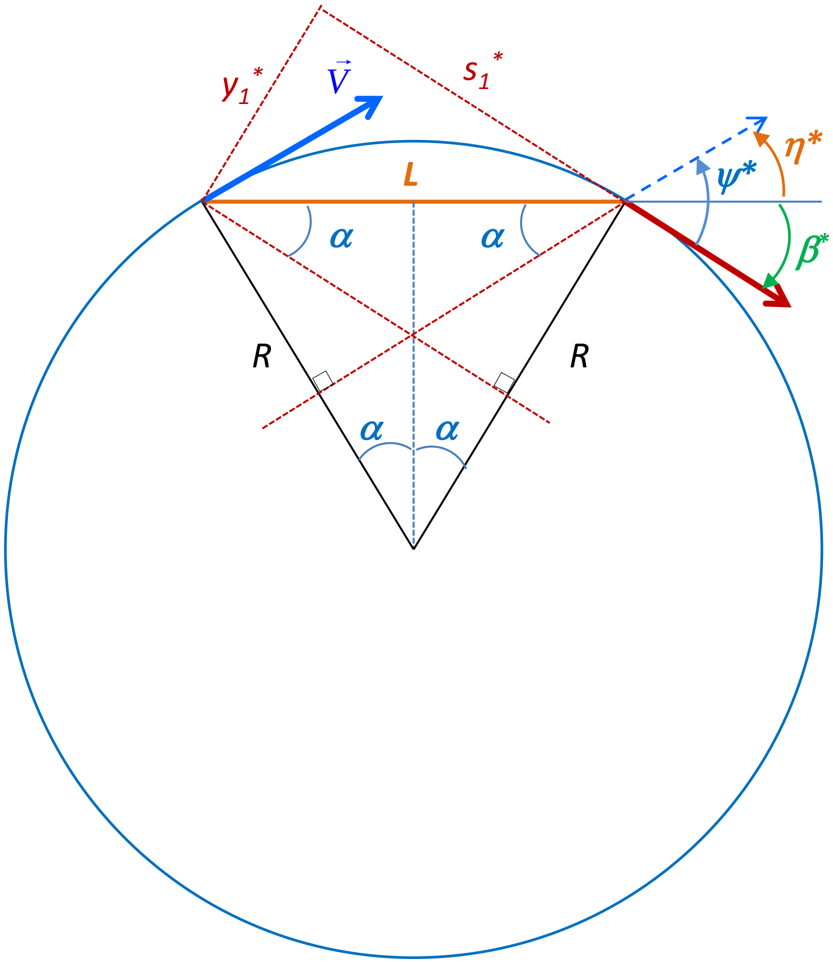
$$y_1^* = R(1-\cos\psi^*)$$

$$\cos\psi^* = 1-\frac{K}{V}(s_1^* + L)$$

$$\sin\eta^* = -\frac{L}{2R}, \quad L < 2R$$

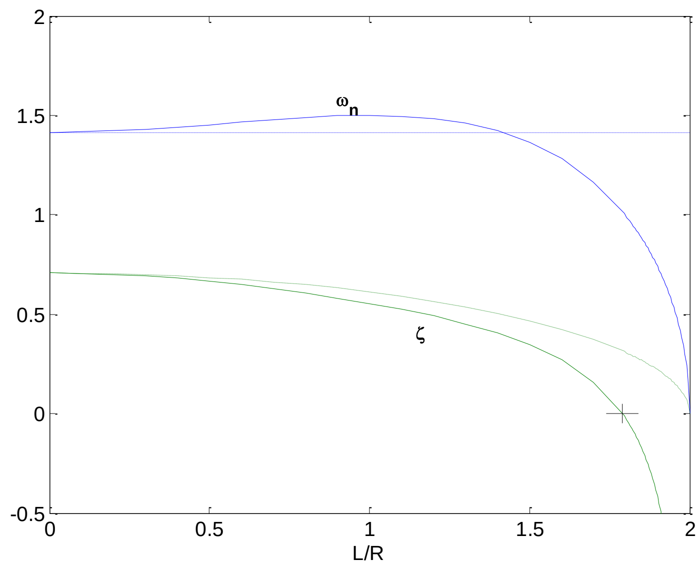
$$\beta^* = \eta^* - \psi^*, \quad \eta^* = -\beta^*, \quad \psi^* = -2\beta^*$$

$$\sin\beta^* = \sin\alpha = \frac{L}{2R},$$



Circular Path Following: Linearized system

- Linearazing the equations of the cross-track error and course error about de e.p. a second order time is obtained with the condition that the vehicle is at a distance *L* of the reference point.



- The linear system is exponentially stable when $0 \leq L/R \leq 1.79$
- The linear system is unstable when $1.8 \leq L/R$

- Dotted curves show the corresponding values obtained by Park et al. 2007

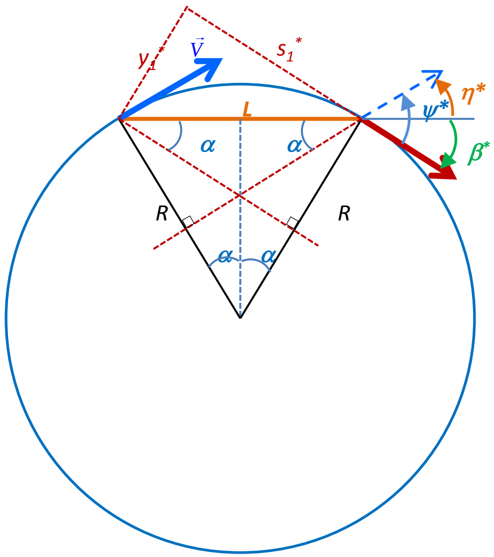
Circular Path Following: Domain of Attractions

Theorem 1. Consider the autonomous system $dx/dt = f(x)$, $x \in R^2$ and let $M \subseteq R^2$ be a compact invariant set for the system with only one equilibrium point in its interior and no equilibrium points on the boundary. Assume that for each initial condition in *M* there is a unique solution, and that *f(x)* has continuous partial derivatives in the interior of *M*. Let *J* denote the Jacobian matrix of the system. Then, if the trace of *J* is negative and the determinant of *J* is positive at the equilibrium point, the domain of attraction is either the set *M* or an open set Ω, whose boundary is a positively invariant periodic orbit. In the latter case, the limit set of the trajectories not in Ω are periodic orbits.

Corollary. Theorem 1 tells us about the behavior when the hyperbolic equilibrium is stable. If the hyperbolic equilibrium point is unstable, then *M* contains at least a limit cycle.

Circular Path Following: Domain of Attractions

- The kynemtic equations can be written as follows



$$\frac{L}{V}\dot{\beta} = \left(\cos\psi + \frac{KL}{V}(1-\cos\beta)\right)\left(\frac{L}{R} - \sin\beta\right) + \sin(\psi+\beta)$$

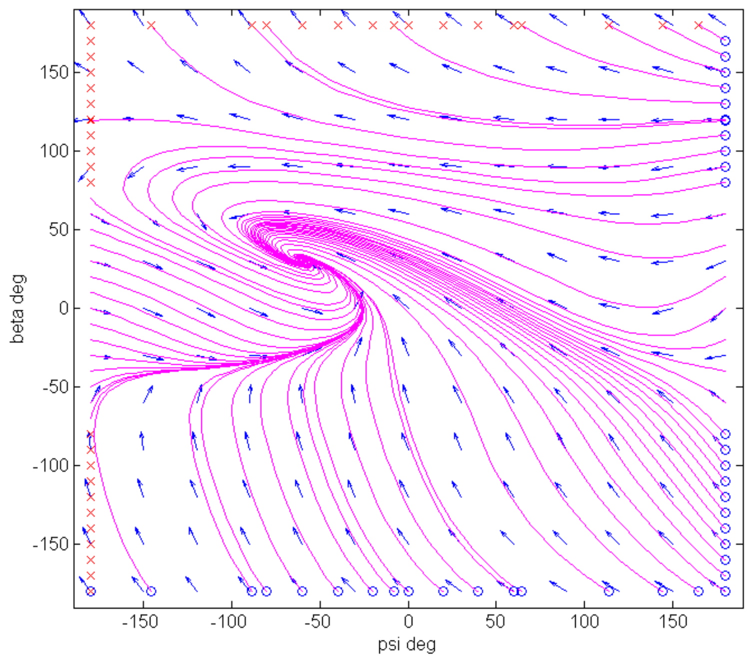
$$\frac{L}{V}\dot{\psi} = \begin{cases} -2\sin(\beta+\psi) - \frac{L}{R}\left(\cos\psi + \frac{KL}{V}(1-\cos\beta)\right), & |\beta+\psi| \le \frac{\pi}{2} \\ -2\mathrm{sign}(\beta+\psi) - \frac{L}{R}\left(\cos\psi + \frac{KL}{V}(1-\cos\beta)\right), & \frac{\pi}{2} < |\beta+\psi| \end{cases}$$

$$\text{domain} \quad Q = \{(\beta,\psi) : \beta,\psi \in [-\pi,\pi]\}$$

- Three different situations are found to the equilibrium point we are analyzing

Circular Path Following: Domain of Attractions

i) *0* ≤ *L/R* ≤ 1.6 The kynematic system is UGAS and ULES

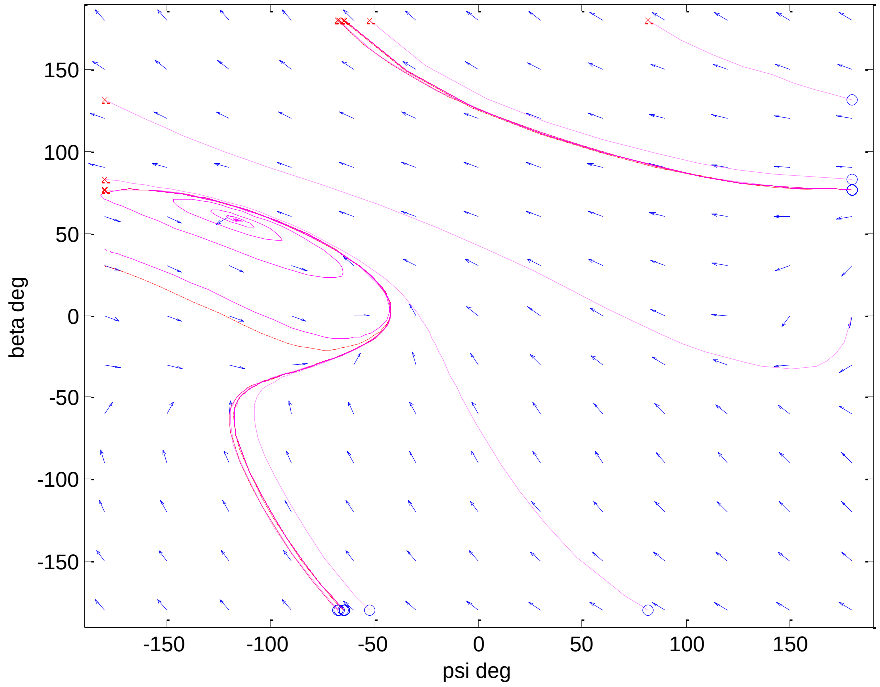


Phase portrait for *L/R* = 1, β^* = *30* deg, ψ^* = *-60* deg.

All trajectories converge to the stationary point. Blue arrows show the flow vector.

Circular Path Following: Domain of Attractions

ii) *1.6 < L/R* ≤ 1.79 The kynematic system is ULAS and ULES and the domain of attraction is a limit cycle.

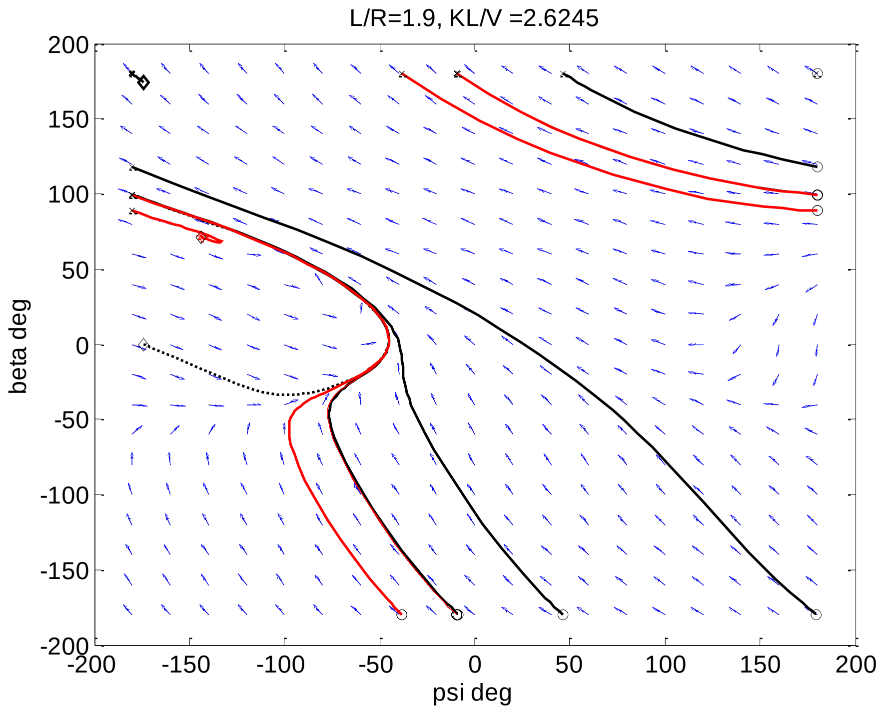


Phase portrait for *L/R* = 1.71. $\beta^* = 58.76$ deg, $\psi^* = -117.52$ deg.

Some trajectories converge to the stationary point and the rest to the limit cycle.

Circular Path Following: Domain of Attractions

iii) *1.7* < *L/R* < 2.0 The equilibrium point is stable and there is a stable limit cycle



Phase portrait for *L/R* = 1.9. β^* = *71.81* deg, ψ^* = *-142.62* deg.

SIMULATION: MODEL

- Kinematic model of the vehicle

$$\dot{x} = V \cos \psi_{V_l} + w_x$$
$$\dot{y} = V \sin \psi_{V_l} + w_y$$
$$\ddot{\psi}_{V_l} * \tau + \dot{\psi}_{V_l} = \dot{\psi}_V$$

w_x, w_y are the components of the wind in the north and east directions, respectively.

The inner loop is modeled as a first order lag with time constant τ.

In all simulations

V = 16 m/s

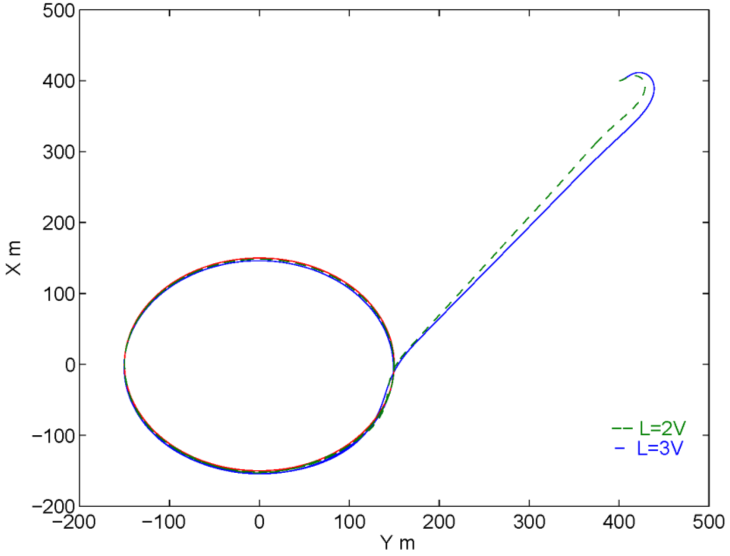
τ *= 1 s*

wind with constant speed of *8 m/s*

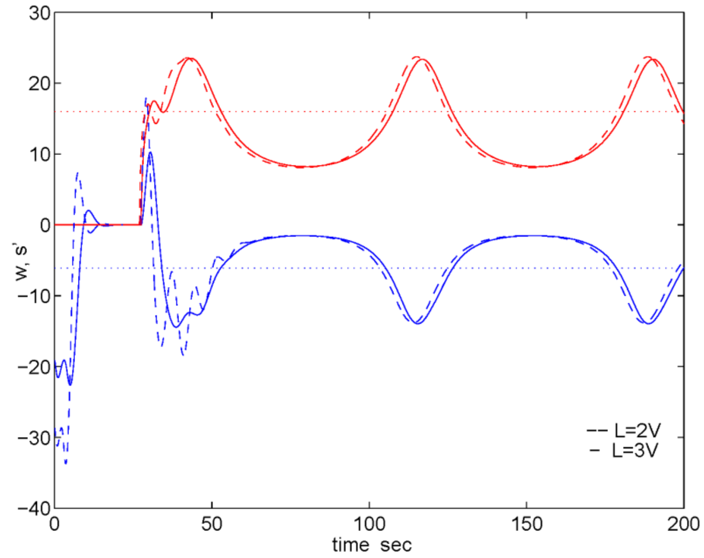
*L = 2*V = 32 m*

*L = 3*V = 48 m*

SIMULATION: CIRCLE



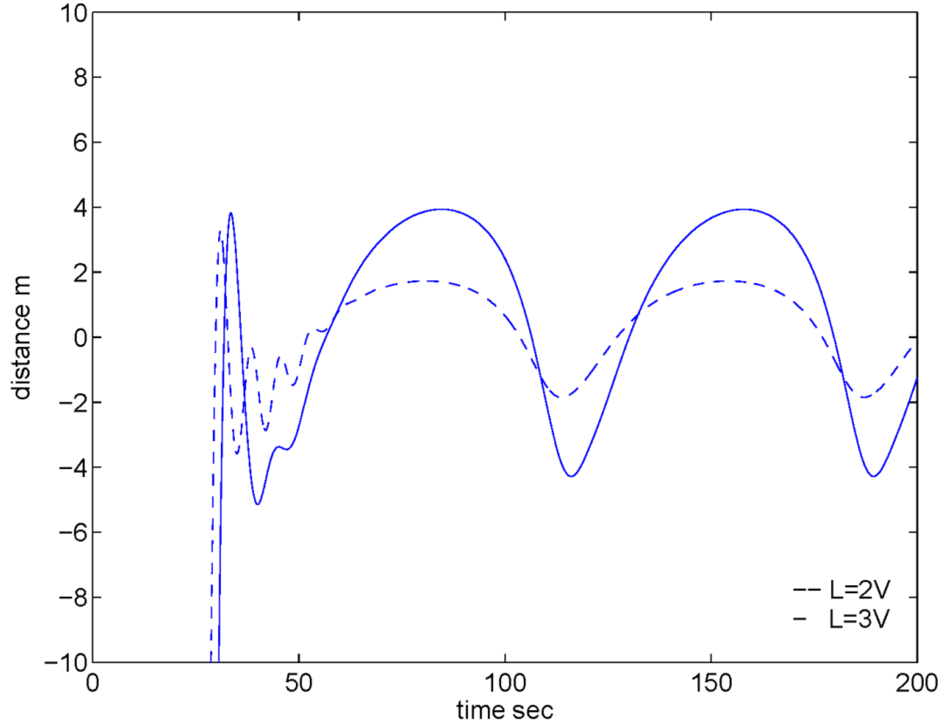
Trajectory of the vehicle (green and blue) and the reference point (red)



Control signals: ψ_V (*deg/s*) in blue, and *ds/dt (m/s)* in red

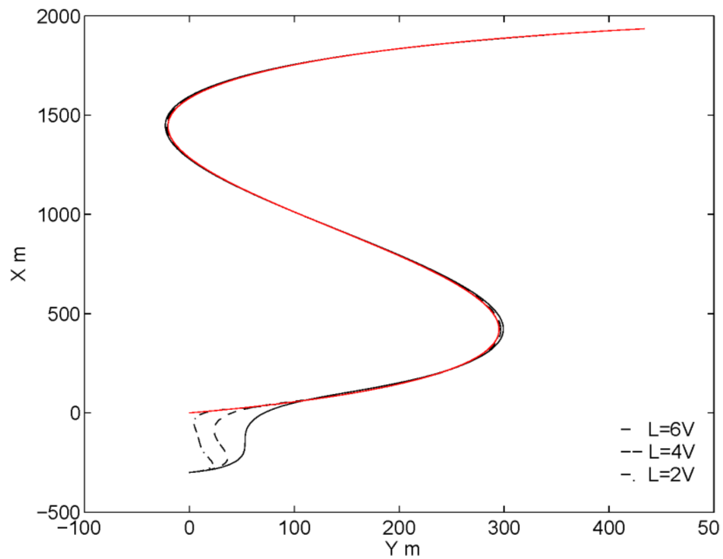
SIMULATION: CIRCLE

Distance of the vehicle to the circle

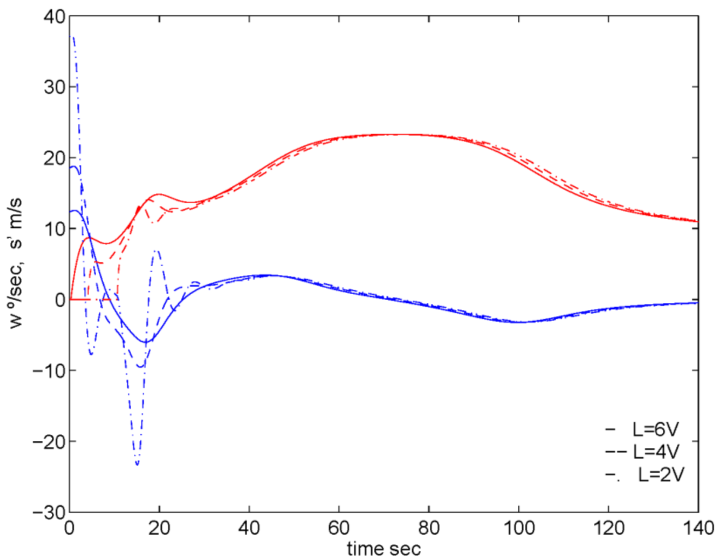


When *L = 3V* the mean following error when the circle has been reaches is 1.0 *m* with standard deviation *1.2 m.*

SIMULATION: Parameterized Curve



Trajectory of the vehicle (black)
and the reference point (red)



Control signals: ψ_V (*deg/s*) in blue, and *ds/dt (m/s)* in red

Maximum separation error at the curves: *0.5 m, 1.5 m, 4 m* for *L = 2V, 4V, 6V.*

Aplications

A Streamlined Nonlinear Path Following Kinematic Controller

Jesús Manuel de la Cruz García
(jmcruz@ucm.es)

Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

UNED, 24 de marzo de 2015