Addressing Unmodelled Path-Following Dynamics via Adaptive Vector Field: a UAV Test Case

ARC course Project

Chepuri Shivani

2018122004

Aim/Objectives

- To understand the path following problem and relevant UAV dynamics
- To understand the use of Vector Fields in UAV (autonomous) path following
- To study the Ideal, Standard and newly proposed Adaptive (Main paper) Vector Field based path following
- To study the relevant control strategies employed
- To present comparisons and analysis
- Simulations and results

Contents/Workflow

Introduction

Model Based Path Following (UAVs) - Major Drawbacks

- **Dependence on the wind knowledge:**
  - In applications like exploration of new areas (MARS rovers, etc) we cannot anticipate the changes in wind.
- **Fidelity/Confidence of the dynamic model used for design**:
  - Unmodeled dynamics can make the system unstable.
  - Not accounting for some variables in the system (possible in linear and non-linear systems)
  - Linearizing a nonlinear system (order reduction)
  - Unmodeled dynamics are not observable by the model, yet they are affected by the controller.

In this paper, adaptive Vector Field control law has been proposed to tackle these two issues.

Need for high performance path-following strategies

Civil and Military UAVs have applications like monitoring, aerial mapping, small cargo deliveries, search and rescue operations etc

They may have to operate in windy environments where wind speeds are 20-50% of the UAV airspeed

The UAV needs to autonomously follow predefined paths at constant height and during climb and descent.

Therefore, UAVs must rely on accurate path- following algorithms which are robust to

- Wind disturbances (wind vector)
- Unmodeled dynamics (course angle dynamics)
- The quality of sensing and control

These are all critical limits to the achievable accuracy

Path Following Techniques

- **Geometric**

  The pure pursuit and line-of-sight guidance laws which make use of a virtual target point where the UAV is directed to

- **Control-theoretic approaches**

  PIDs, linear quadratic control, sliding-mode control, model predictive control, adaptive control and their variants

- **The Vector Field (VF)** approach sits in between these two classes, by combining geometric reasoning with a sliding - mode technique

Path Following Problem - Formulation

Given a path, initial location p = (x, y) of the aerial vehicle along with its heading angle (ψ).

The path following problem is to determine the commanded heading angle (ψd) for the vehicle such that by following (ψd), the UAV accurately tracks the path as the mission progresses.

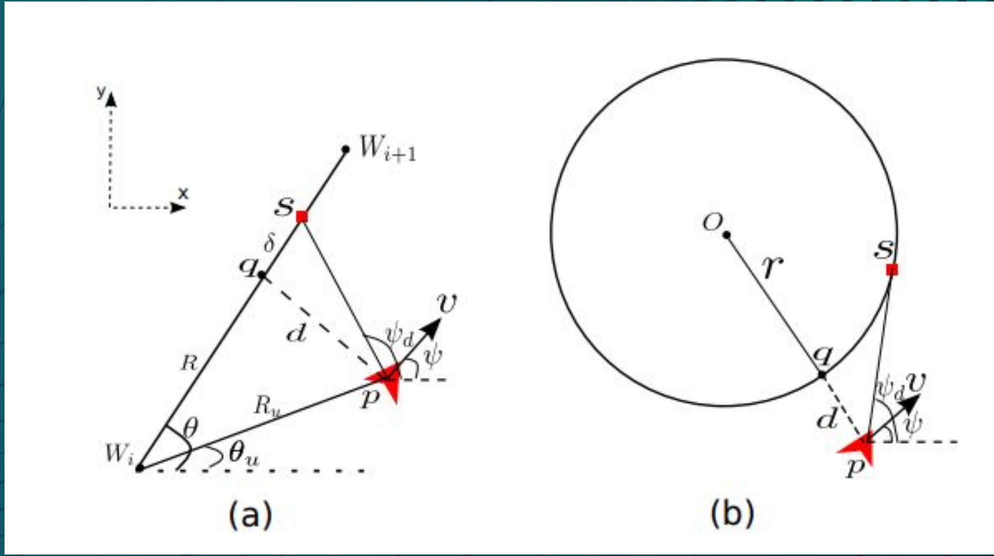


Fig 1

(a) The UAV has to follow a straight line path

(b) The UAV has to follow a loiter or orbit of radius r

Path Following Problem - Formulation

- **Straight Line Paths and Circular Orbit Paths:**

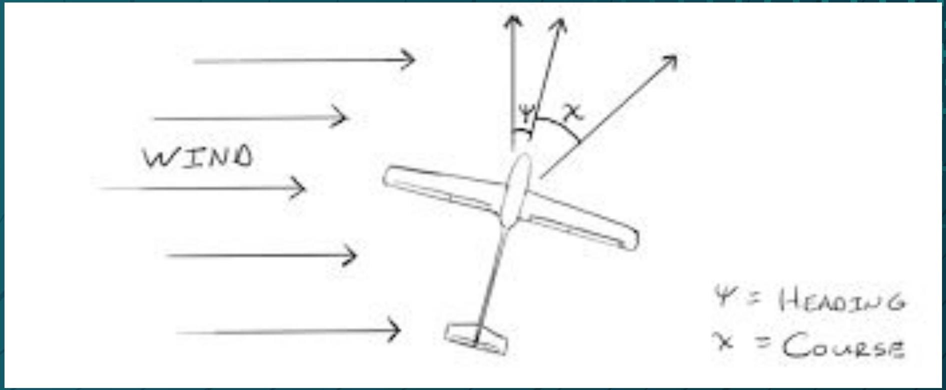
  A set of waypoints. The angle formed by waypoints W_i and W_i+1 is called the **line-of-sight (LOS)** angle.
- Assume that the UAV is at a distance 'd' from the path. This distance d is called as **crosstrack error**.
- **Objective of the problem:**
  - Minimize the cross-track error
  - Ensure that ψ is same as the LOS angle θ.
  - That is, make $\mathbf{d} \rightarrow \mathbf{0}$ and $|\psi - \theta| \rightarrow \mathbf{0}$, where |.| represents the absolute value as the mission time $\mathbf{t} \rightarrow \mathbf{inf}$.
  - The objective is to be on the path rather than at a certain point at a particular time, therefore the time dependence of the problem is removed.

Heading and Course Angles

Heading (yaw) Angle (ψ): The direction the nose of the UAV is pointing

Course angle (χ):

- The direction the UAV is actually moving.
- It is the angle between true north & projection of **Vg** on the horizontal plane (i_b, j_b) (in wind triangle)
- **χ** is the control variable for the guidance logic



The difference between course and heading is called the crab angle, or **side-slip angle**.

Eg. A helicopter pointing east (heading = 90 degrees) and banking left is moving north (course = 0 degrees).

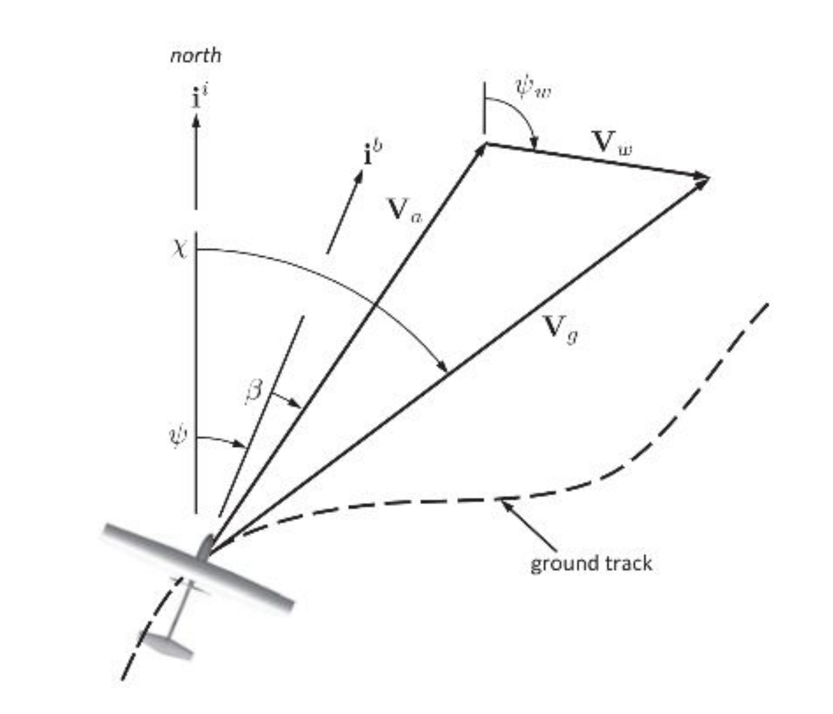
UAV Dynamics - The wind triangle

Gives the relation between Airspeed Va , ground speed Vg , and wind speed Vw

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w.$$

Wind is expressed as a sum of its dynamical (turbulence) part, static (const amplitude and direction) and time varying perturbations (sinusoidal)

$$\mathbf{V}_w = \mathbf{V}_{w,s} + \mathbf{V}_{w,d} + \mathbf{V}_{w,v}$$



UAV Course angle Dynamics

From [2], we can say longitudinal and lateral dynamics of a UAV are decoupled.

The linearized lateral dynamics around the trim equilibrium are the following transfer functions: (lateral dynamics are relevant to path follow)

$$\text{Roll angle} \qquad \phi(s) = \frac{a_{\phi_2}}{s(s + a_{\phi_1})}\left(\delta_a(s) + \tilde{d}_\phi(s)\right)$$

$$\text{Course angle} \qquad \chi(s) = \frac{g}{V_g s}\left(\phi(s) + \tilde{d}_\chi(s)\right)$$

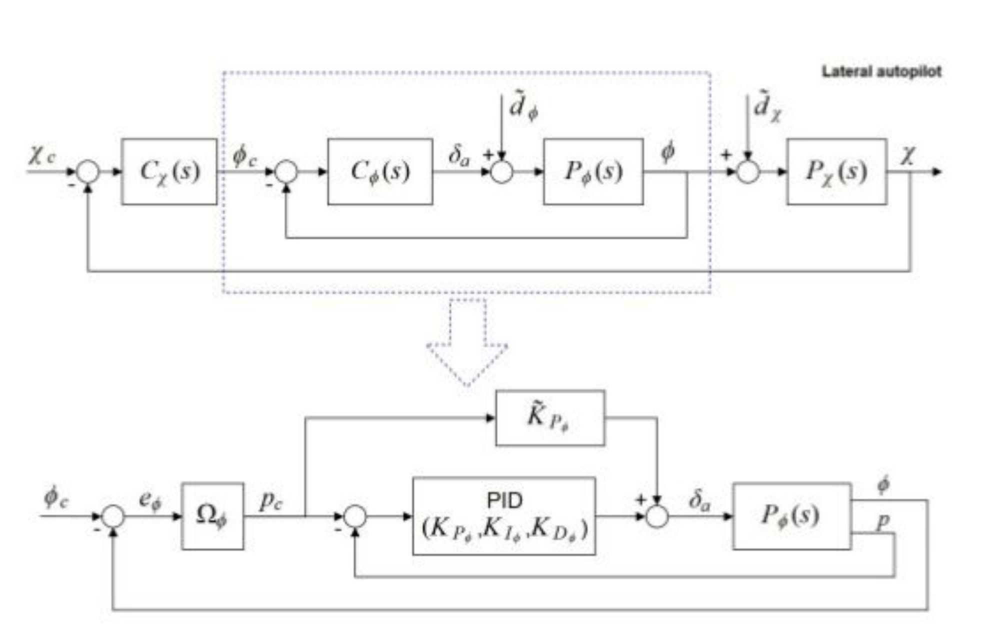
δa is aileron command (roll control, changes lift)

d˜φ and d˜χ are disturbances coming from cross effects of neglected dynamics

aφ1 , aφ2 are some constants

Unmodelled course angle dynamics - ArduPilot - Lateral Autopilot

- Open source Autopilot software for drones and other autonomous systems
- To quantify the distance between the first-order dynamics and the fourth-order dynamics (next slide)
- **3** nested loops for roll rate **p**, roll angle **φ**, outermost for course angle **χ**.



Unmodelled course angle dynamics - ArduPilot

Done to obtain 1st or 2nd order dynamics so PID can be used for low level design (attitude and position control)

- We get the following transfer functions: 2nd is obtained by neglecting d˜χ.
- 3rd is the final fourth-order course angle dynamics.
- 4th is obtained by neglecting higher order terms in 3rd. *(figure in previous slide)*

$$\frac{\phi(s)}{\phi_c(s)} = \frac{2017.8}{(s^2 + 8.467s + 44.88)(s + 45)}$$

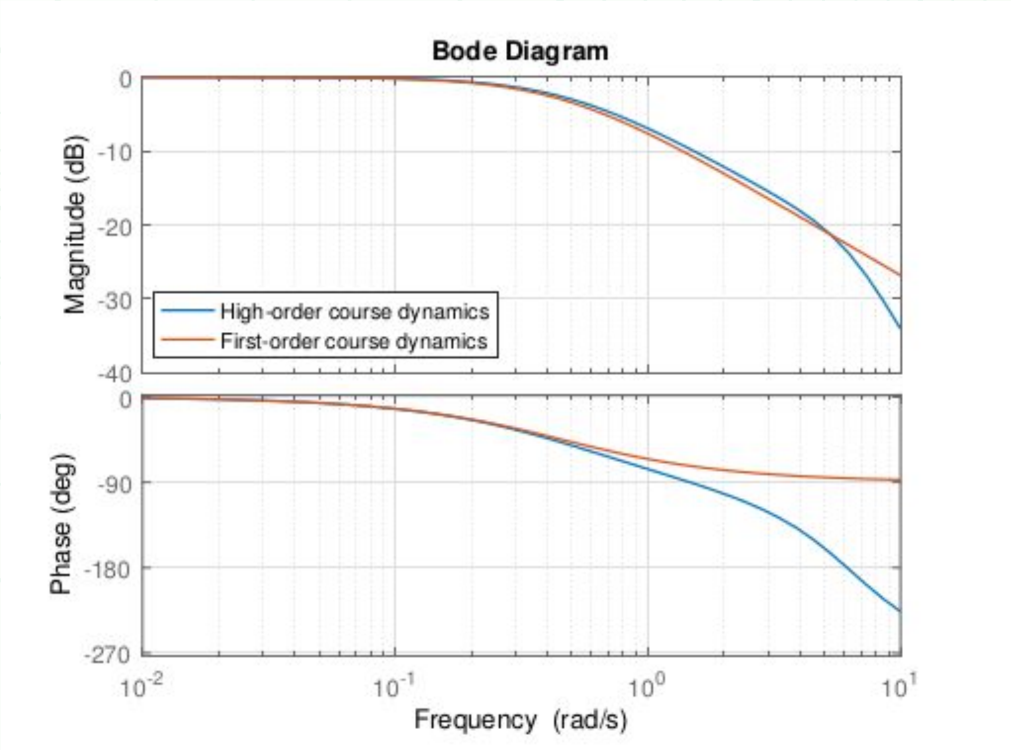
$$\frac{\chi(s)}{\phi_c(s)} = \frac{\phi(s)}{\phi_c(s)} \frac{g}{V_g s} = \frac{0.654\phi(s)}{s\phi_c(s)}.$$

$$\frac{\chi(s)}{\chi_c(s)} = \frac{923.72}{(s + 0.51)(s^2 + 7.97s + 40.38)(s + 44.99)}$$

$$\left(\frac{\chi(s)}{\chi_c(s)}\right)_{1st} = \frac{\frac{C_\chi g}{sV_g}}{1 + \frac{C_\chi g}{sV_g}} = \frac{\alpha_\chi}{s + \alpha_\chi} \Rightarrow \dot{\chi} = \alpha_\chi(\chi_c - \chi)$$

Effects of Neglecting cross-effects $\tilde{d}_\varphi$ and $\tilde{d}_\chi$:

- **-3dB bandwidth** of (3rd eq) is 11% larger than the bandwidth of (4th eq) Therefore, operating range of the UAV is affected.
- So, first-order approximation necessarily creates unmodeled dynamics
- Comparison of Bode Diagrams of 4th order and 1st order dynamics respectively



Vector Fields

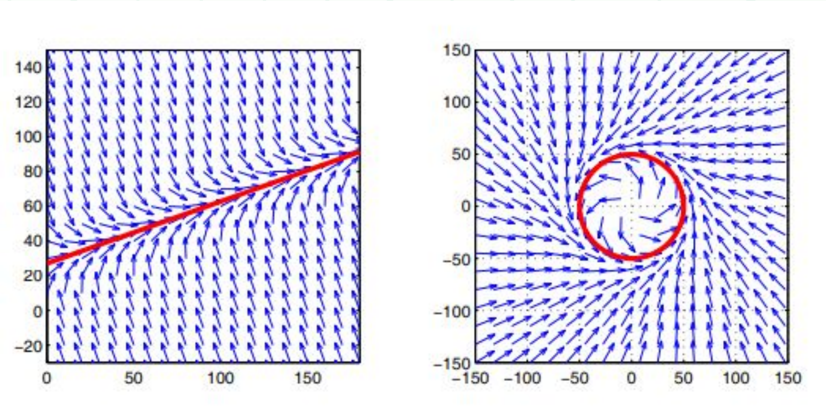
The notion of vector fields is similar to that of potential Fields

They do not necessarily represent the gradient of a potential but simply indicate a desired direction of travel.

We generate VF such that given any initial point, we want the UAV to converge to a given path.

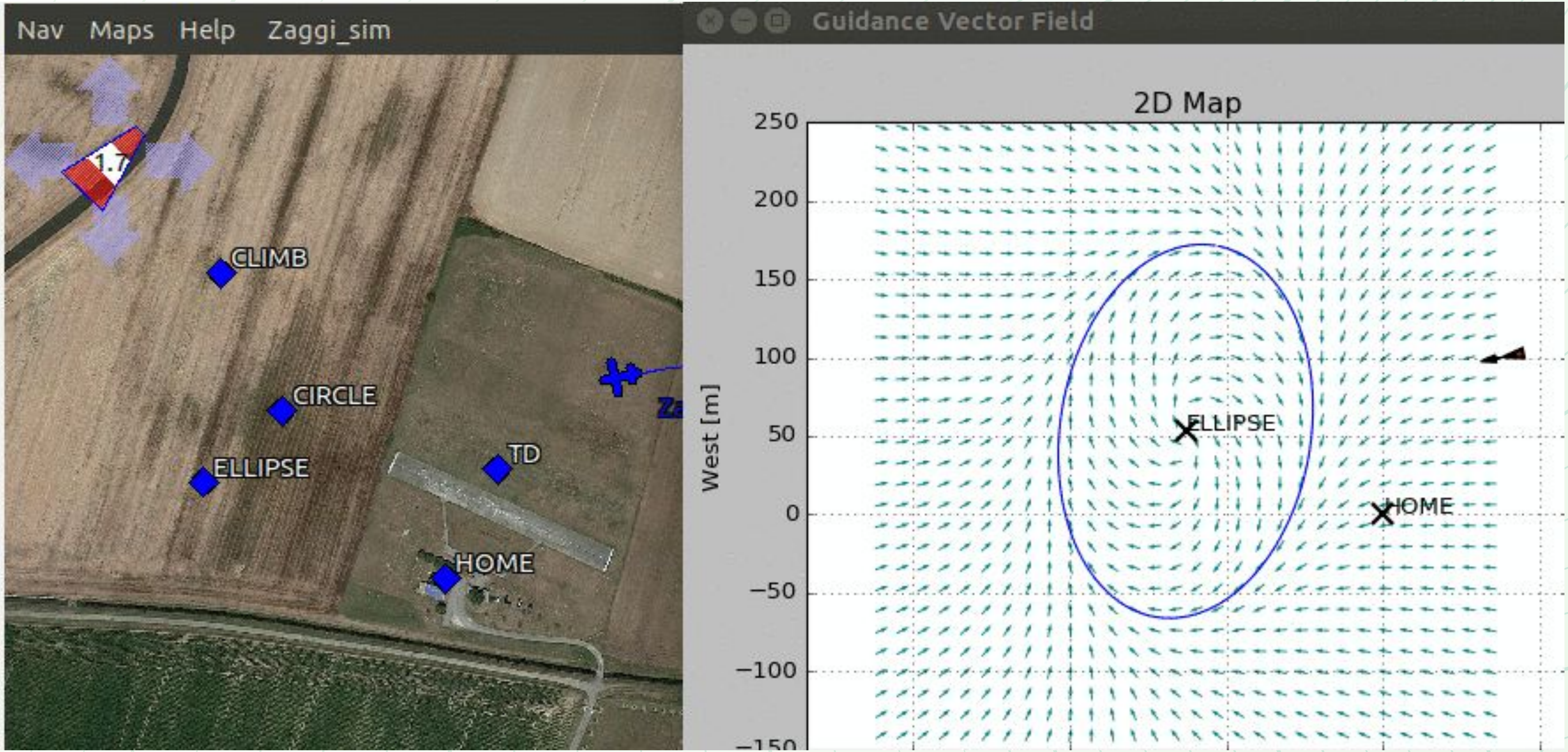
Most commonly, **straight line and circular orbit paths**. (see sim results)

The vectors of the fields provide course commands to guide the MAV toward the desired path.



Working of a guidance VF

[4]



Tackling Unmodeled dynamics

Standard VF based path following - assumes known constant wind and uses first-order course dynamics

Ideal VF based path following - assumes dynamic, static, time varying wind and uses first-order course dynamics

Standard VF based Path Following

Assumptions: Altitude and Airspeed (Va) are held constant (or nearly so) by the control of the longitudinal dynamics

Applications: collision avoidance and target tracking

The method calculates a vector field around the path to be tracked.

Equations of Motion:

- Expressed in terms of ground speed and course
- Independent of the wind velocity

$$\dot{x} = V_g \cos \chi$$
$$\dot{y} = V_g \sin \chi$$

Dynamics: 'alpha' is a known positive constant that characterizes the speed of response of course hold autopilot loop

$$\dot{\chi} = \alpha (\chi^c - \chi)$$

Straight Line Path Following

- Far away from the waypoint path, the vector field is directed with an angle X_inf from the perpendicular to the path.

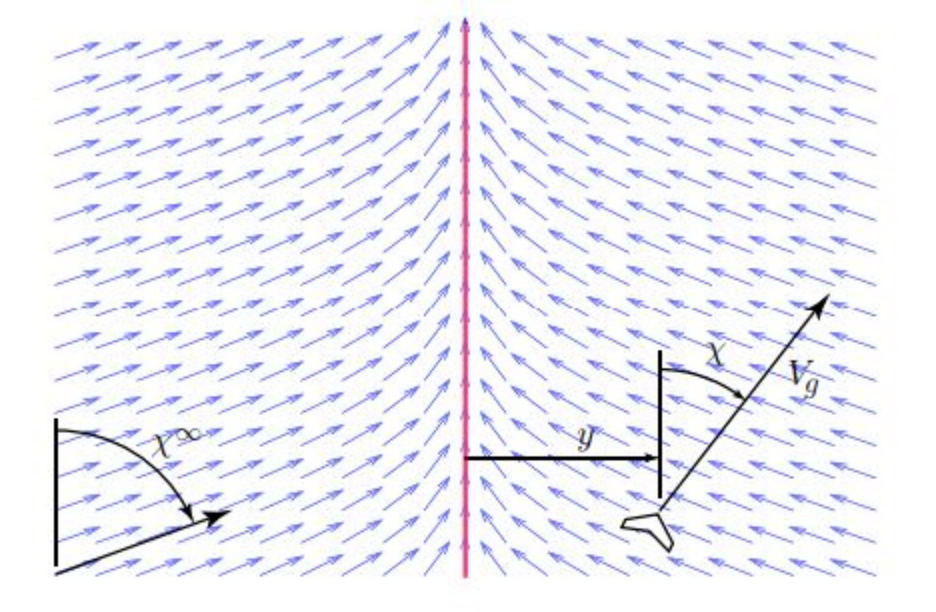
  Let y = e_py (cross track error from point 'p' to the desired course)

  The reference course VF is

$$\chi^d(y) = -\chi^\infty \frac{2}{\pi} \tan^{-1}(ky),$$

If χ^∞ is restricted to be in the range $\chi^\infty \in (0, \frac{\pi}{2}]$ then clearly

$$-\frac{\pi}{2} < \chi^\infty \frac{2}{\pi} \tan^{-1}(ky) < \frac{\pi}{2}$$



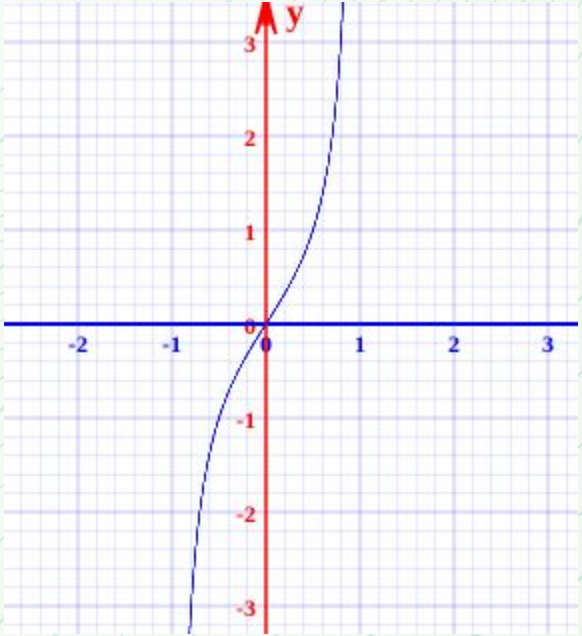
Straight Line Path Following

- x = (2/pi)*atan(y) graph -> converges to a straight line (x=1 if x>0 and x= -1 if x<0)
- Consider the Lyapunov function $W_1(y) = \frac{1}{2}y^2$
- Taking the Lie derivative,

$$\begin{aligned}\dot{W}_1 &= V_g y \sin(\chi^d(y)) \\ &= -V_g y \sin\left(\chi^\infty \frac{2}{\pi} \tan^{-1}(ky)\right),\end{aligned}$$

which is less than zero for y != 0.

That is, if X = X_d (y), then y -> 0 asymptotically.



(simulated plot)

Sliding mode control - Finite Time Stability

Consider the following set

$$S = \{(y, \chi) : \chi = \chi^d(y)\}$$

We want to make sure that the system trajectory reaches S in a finite time. Let

$$\tilde{\chi} \triangleq \chi - \chi^d(y)$$

Let $W_2 = \frac{1}{2}\tilde{\chi}^2$ and take the derivative to obtain

$$\begin{aligned}
\dot{W}_2 &= \tilde{\chi}\dot{\tilde{\chi}} \\
&= \tilde{\chi}\left(\dot{\chi} - \dot{\chi}^d(y)\right) \\
&= \tilde{\chi}\left(\alpha(\chi^c - \chi) + \chi^\infty \frac{2}{\pi}\frac{k}{1+(ky)^2} V_g \sin\chi\right).
\end{aligned}$$

Sliding mode control - Finite Time Stability

If we choose the control signal as

$$\chi^c = \chi - \frac{1}{\alpha}\chi^\infty \frac{2}{\pi}\frac{k}{1+(ky)^2}V_g \sin\chi - \frac{\kappa}{\alpha}\text{sign}(\tilde{\chi}),$$

For _k > 0, We get $\dot{W}_2 \leq -\kappa\,|\tilde{\chi}|$ implies, X_tilda -> 0 in finite time

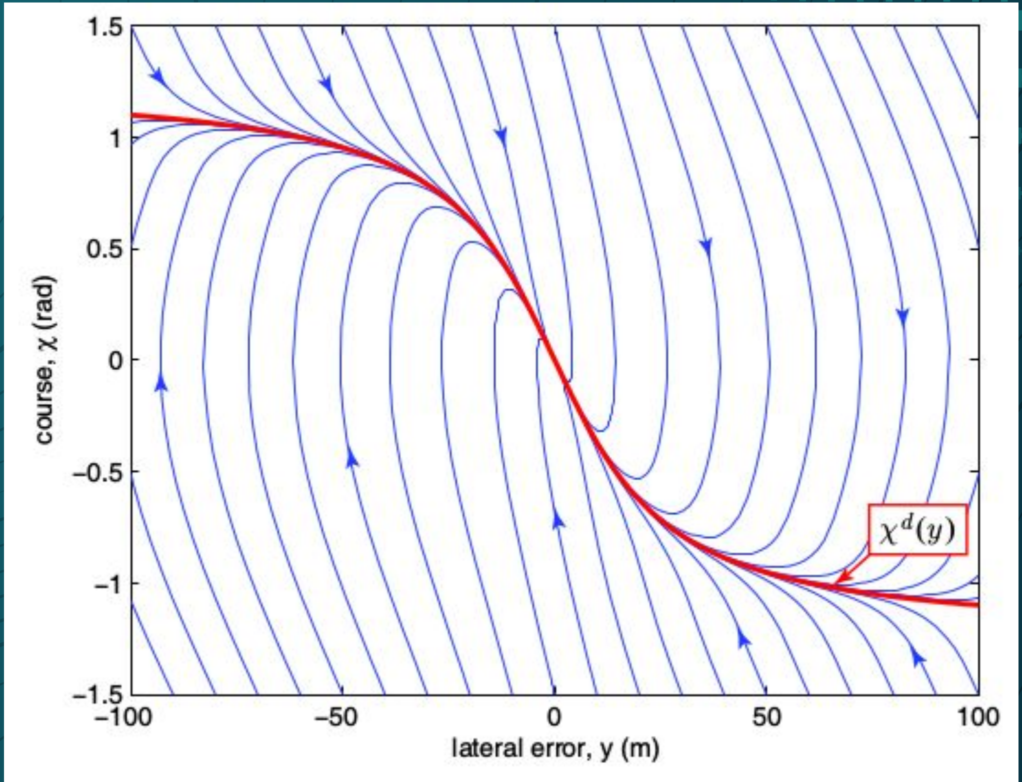
To avoid chattering, we can use a saturation function in place of sign(X_tilda)

Results - Sliding Surface

The shape of the sliding surface and system trajectory onto the sliding surface are influenced by the control parameters k, _k, e (in saturation function), X_inf.

State trajectories onto the sliding surface X_d (y) for various initial conditions.

(matlab)



Adaptive Vector Field based Control Law

Tackles unmodeled dynamics with almost zero knowledge of wind

Adaptive VF Path Following - Straight Line

We adapt Vg depending on the cross track error (y or e_py)

We will consider $\hat{V}$g in place of **Vg** , where $\hat{V}$g is adapted by an auxiliary differential equation.

We only consider wind as constant at t = 0 and later is unknown.

The estimation dynamics for a straight-line path is,

$$\dot{\hat{V}}_g = \Gamma_{sl}\mu_{sl}\tilde{\chi}\chi_\infty\beta_s\frac{2}{\pi}\sin(\chi-\chi_q) + F_{sl} - \sigma_{sl}\Gamma_{sl}\hat{V}_g$$

$$F_{sl} = \frac{\partial\hat{V}_g}{\partial\chi}\left[-\chi_\infty\frac{2}{\pi}\beta_s\hat{V}_g\sin(\chi-\chi_q) - \kappa_{sl}\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon_{sl}}\right)\right]$$

$$\frac{\partial\hat{V}_g}{\partial\chi} \simeq W_s\left[\sin(\psi_{w,s}-\chi) + \left(V_a^2 - W_s^2\sin^2(\psi_{w,s}-\chi)\right)^{-\frac{1}{2}} \cdot\sin(\psi_{w,s}-\chi)\cos(\psi_{w,s}-\chi)\right]$$

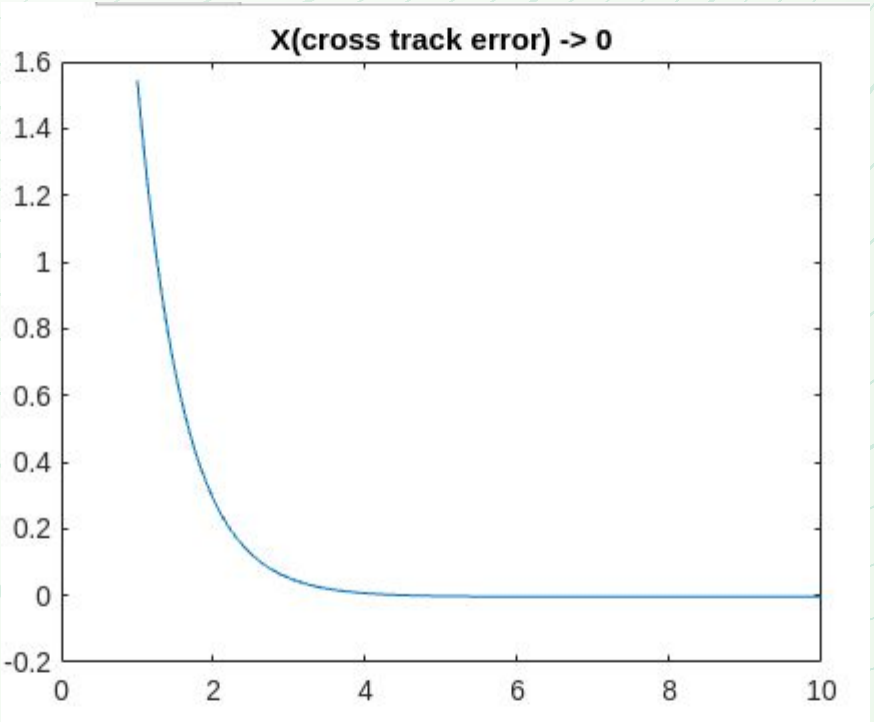
Γ_sl is the estimator gain
μ_sl is a weighting term
σ_sl adds damping action

The feedforward term F_sl accounts for the variation of V_g with respect to the course angle χ

$$W_s = \|\mathbf{V}_{w,s}\|$$

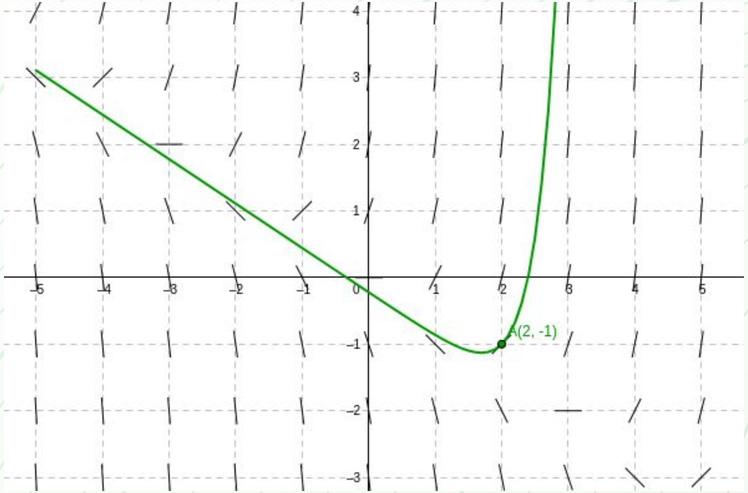
Simulation and Results

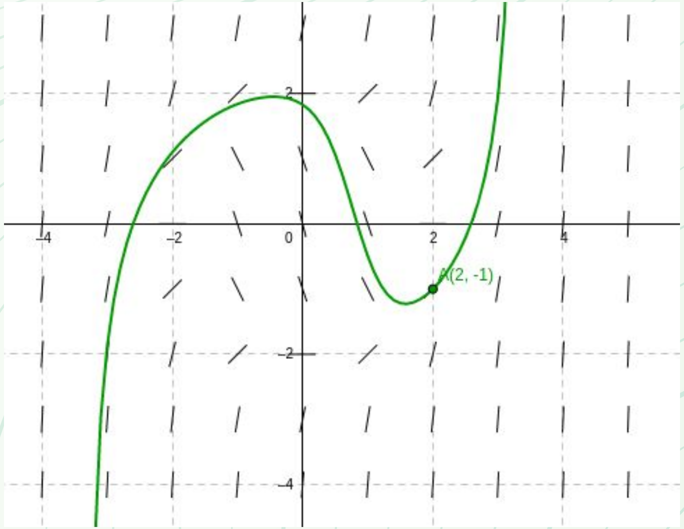
```
%dynamics - integration
for i=1:length(T)
    xd = xd_fun(k,y,i);
    x_tilda = x_err(xd, x,i);
    xc = x_c(x,y,x_inf,alpha,k,K,e,i,Va,V_ws);
    x_s = x_s + Ts*(alpha*(xc - x_s));
    vg = Vg(Va(i),V_ws(i));
    y_s = y_s + Ts*(vg*sin(x(i)));
    if(i>1)
    x(i) = x_s;
    y(i) = y_s;
    xc_i(i) = xc;
    end

end
```



Simulation and Results

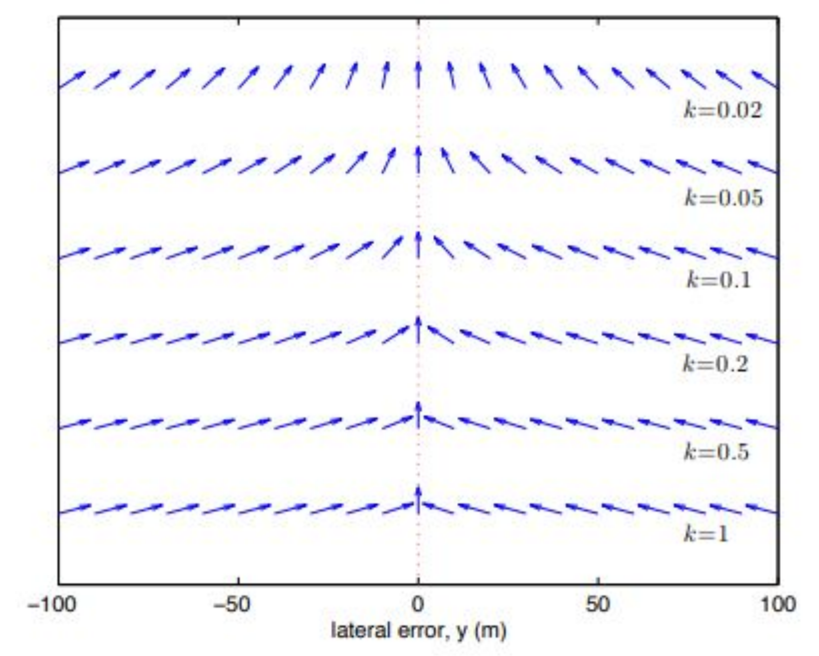
Vector Fields - straight line + circular orbits (plotted using geogebra.org)





Results- Changing k parameter, changing transition timing

- k is a positive constant that influences the rate of the transition from X_inf to zero.
- Large k results in short, abrupt transitions in the desired course.
- small k results in long, smooth transitions in the desired course.
- (matlab)



AVF Stability Analysis

Consider the Lyapunov function $\mathcal{V} = \mathcal{V}_1 + \rho\mathcal{V}_2 + \frac{1}{2}\Gamma_{sl}^{-1}\Theta^2$, with $\mathcal{V}_1 = \frac{1}{2}e_{py}^2$, $\mathcal{V}_2 = \frac{1}{2}\tilde{\chi}^2$ whose derivative is

$$\dot{\mathcal{V}} = \dot{\mathcal{V}}_1 + \rho\dot{\mathcal{V}}_2 + \Gamma_{sl}^{-1}\Theta\dot{\Theta}$$

Θ = V̂g – Vg is the estimation error, ρ is the positive user-designed weight.

From slide 21, using V̂g instead of Vg and from V̂g_dot expression, we get a V_dot expression and using modulus & other conditions we can upper bound this

$$\dot{\mathcal{V}} \leq \dot{\mathcal{V}}_1 - \rho\zeta_{sl}\tilde{\chi}^2 - \frac{\sigma_{sl}}{2}\Theta^2 + \frac{\sigma_{sl}(V_g + \dot{V}_g\Gamma_{sl}^{-1}\sigma_{sl}^{-1})^2}{2}.$$

Since we assume that the wind changes in a slowly time-varying fashion, the magnitude of V̇ g will be bounded. V g is bounded.

AVF Stability Analysis

Using the definition of the Lyapunov function V and the analysis of V̇ 1 carried out

$$\dot{\mathcal{V}} \leq -\kappa\mathcal{V} - (\bar{\kappa} - \kappa)\mathcal{V} + C$$

where $\zeta_{sl} \geq \frac{1}{2}\sigma_{sl}\Gamma_{sl}$, $0 < \kappa < \bar{\kappa}$, $\bar{\kappa} = \sigma_{sl}\Gamma_{sl}$, and $C = \frac{\sigma_{sl}\bar{\Delta}_V^2}{2}$,

Let B = C/(κ̄−κ), a scalar, then we have

$$\left\| \left[e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_{sl}^{-1/2}\Theta \right] \right\|^2 \leq \max\{\mathcal{V}(0), B\}$$

e_py , χ̃, Θ converge inside a compact set and stay bounded

=> UUB boundedness. In the presence of disturbances and unmodelled dynamics asymptotic Lyapunov stability cannot be guaranteed

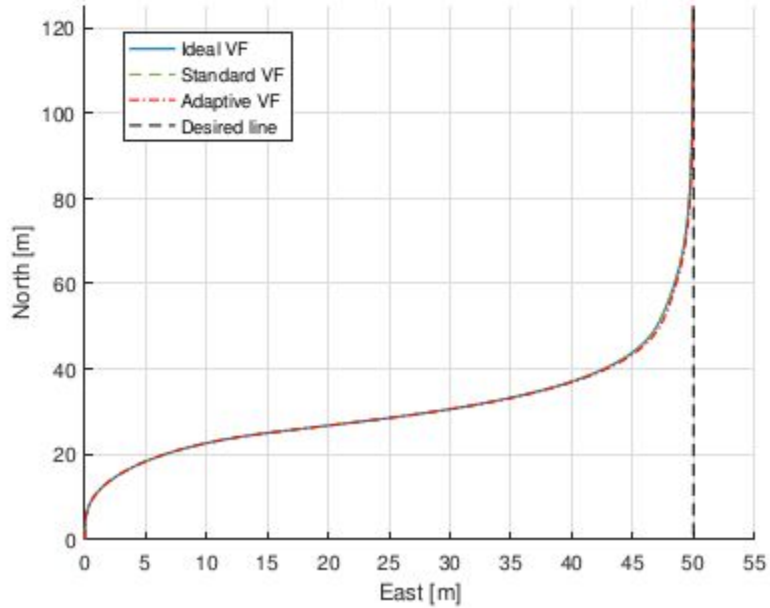
Hardware simulations results and comparison from the paper

All algorithms have almost same transient performance with priori wind knowledge, VF is slightly better

RMS errors

Straight-line path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	23.53	23.52	23.53
#2	26.24	26.20	26.24
#3	26.63	26.58	26.62
#4	27.32	27.08	27.13



Algorithm Evaluation - Inferences from Main paper

- With perfect knowledge of the wind and assuming simplified first-order dynamics, the ideal VF achieves zero steady-state error in all wind conditions

- In the absence of wind, or with only constant wind also the standard and the adaptive VF can achieve zero steady-state-error

- The adaptive VF outperforms the standard VF in Scenarios (error reduction > 20% for the straight line and > 50% for the orbit), when unmodeled wind components cannot be accounted by the standard VF.

References

[1] R. W. Beard and T. W. McLain, Small unmanned aircraft: Theory and practice. Princeton University Press, 2012.

[2] Main paper: https://ieeexplore.ieee.org/document/8747462

[3] https://ieeexplore.ieee.org/document/6669680 - An evaluation of UAV path following algorithms

[4]https://wiki.paparazziuav.org/wiki/Module/guidance_vector_field

[5] https://arxiv.org/pdf/1610.04391.pdf - Vector Field path following

Thank you..

Circular Orbit Path Following - Standard VF

Circular Orbit Path Following - Adaptive VF

Adaptive VF Path Following - Circular Orbit

We adapt Vg depending on the cross track error (y or e_py)

We will consider $\hat{V}$g in place of **Vg** , where $\hat{V}$g is adapted by an auxiliary differential equation.

We only consider wind as constant at t = 0 and later is unknown.

The estimation dynamics for a straight-line path is,

$$\dot{\hat{V}}_g = \Gamma_{sl}\mu_{sl}\tilde{\chi}\chi_\infty\beta_s\frac{2}{\pi}\sin(\chi - \chi_q) + F_{sl} - \sigma_{sl}\Gamma_{sl}\hat{V}_g$$

$$F_{sl} = \frac{\partial \hat{V}_g}{\partial \chi}\left[-\chi_\infty\frac{2}{\pi}\beta_s\hat{V}_g\sin(\chi - \chi_q) - \kappa_{sl}\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon_{sl}}\right)\right]$$

$$\frac{\partial \hat{V}_g}{\partial \chi} \simeq W_s\left[\sin(\psi_{w,s} - \chi) + \left(V_a^2 - W_s^2\sin^2(\psi_{w,s} - \chi)\right)^{-\frac{1}{2}} \cdot \sin(\psi_{w,s} - \chi)\cos(\psi_{w,s} - \chi)\right]$$

Γ_sl is the estimator gain
μ_sl is a weighting term
σ_sl adds damping action

The feedforward term F_sl accounts for the variation of V_g with respect to the course angle χ

$$W_s = \|\mathbf{V}_{w,s}\|$$

Finite Time Stability with sliding mode control - Standard VF