Flight Management Systems

*Flight Management Systems*

# Aircraft guidance

*Joan Vila Carbó*
*Universitat Politècnica de Valencia (UPV)*

# Aircraft guidance

## Bibliography


- **Randy Walter.** The Avionics Handbook. Chap. 15 - “Flight Management Systems”. *Ed. Cary R. Spitzer Boca Raton, CRC Press LLC. 2001*.
- **Maarten Uijt de Haag.** EE6900 Flight Management Systems. Ohio University course.
  - http://www.ohio.edu/people/uijtdeha/ee6900-spring-semester-2014.html
- **Ducard GJJ.** Fault-tolerant Flight Control and Guidance Systems. *Ed. Springer. Series: Advances in Industrial Control. 2009.*
- **Mangal Kothari.** Algorithms for Motion Planning and Target Capturing. *Ph D. Thesis Univ. Leicester. 2010.*
- **Eurocontrol.** User Manual for the Base of Aircraft Data (BADA) Rev. 3.9. EEC Technical/Scientific Report No. 11/03/08-08.
- **The Boeing Company.** 787-8 Flight Crew Operations Manual.

# Aircraft guidance

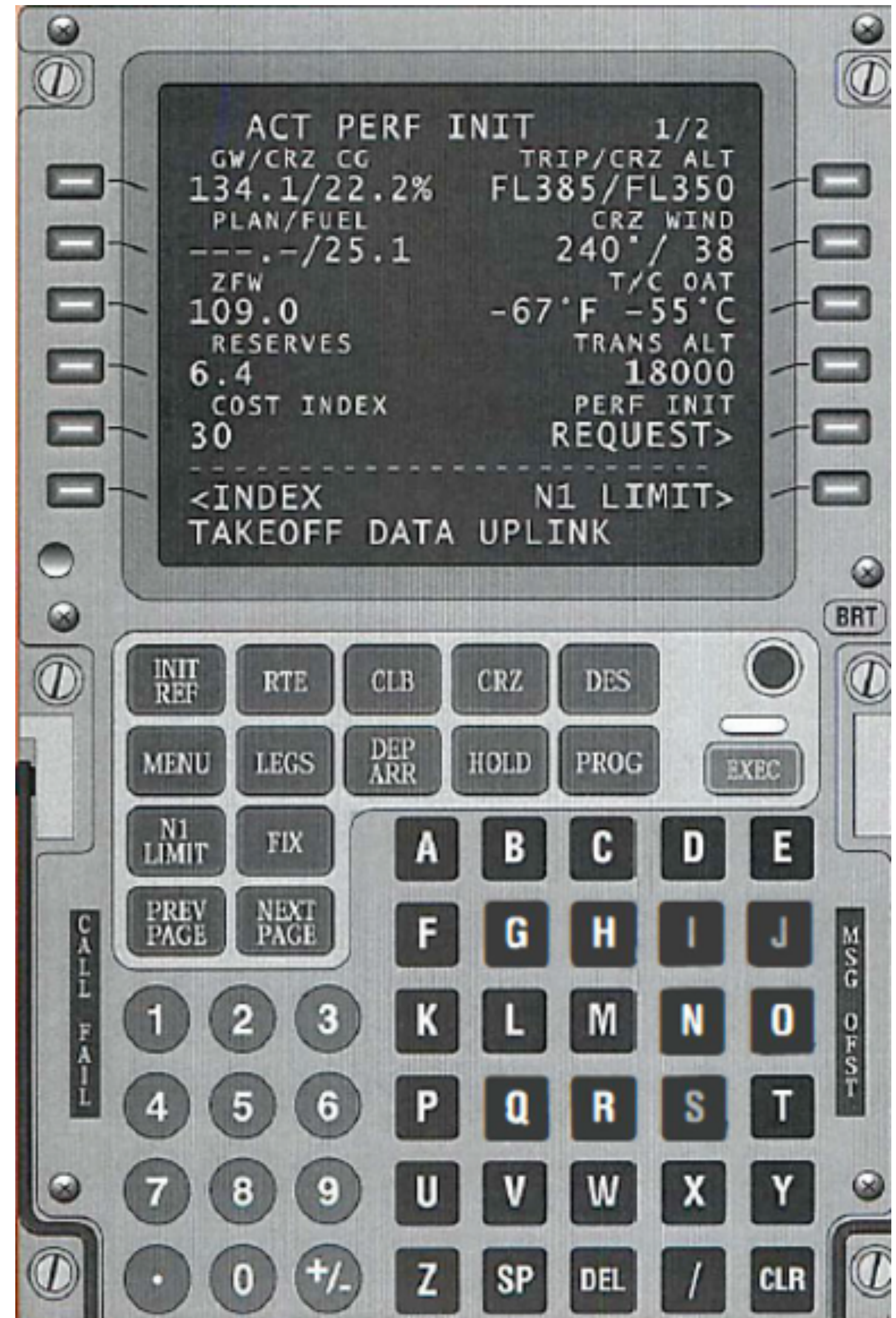
Flight Management Systems

- **Introduction**
- **Lateral guidance**
- **Vertical guidance**

# Introduction

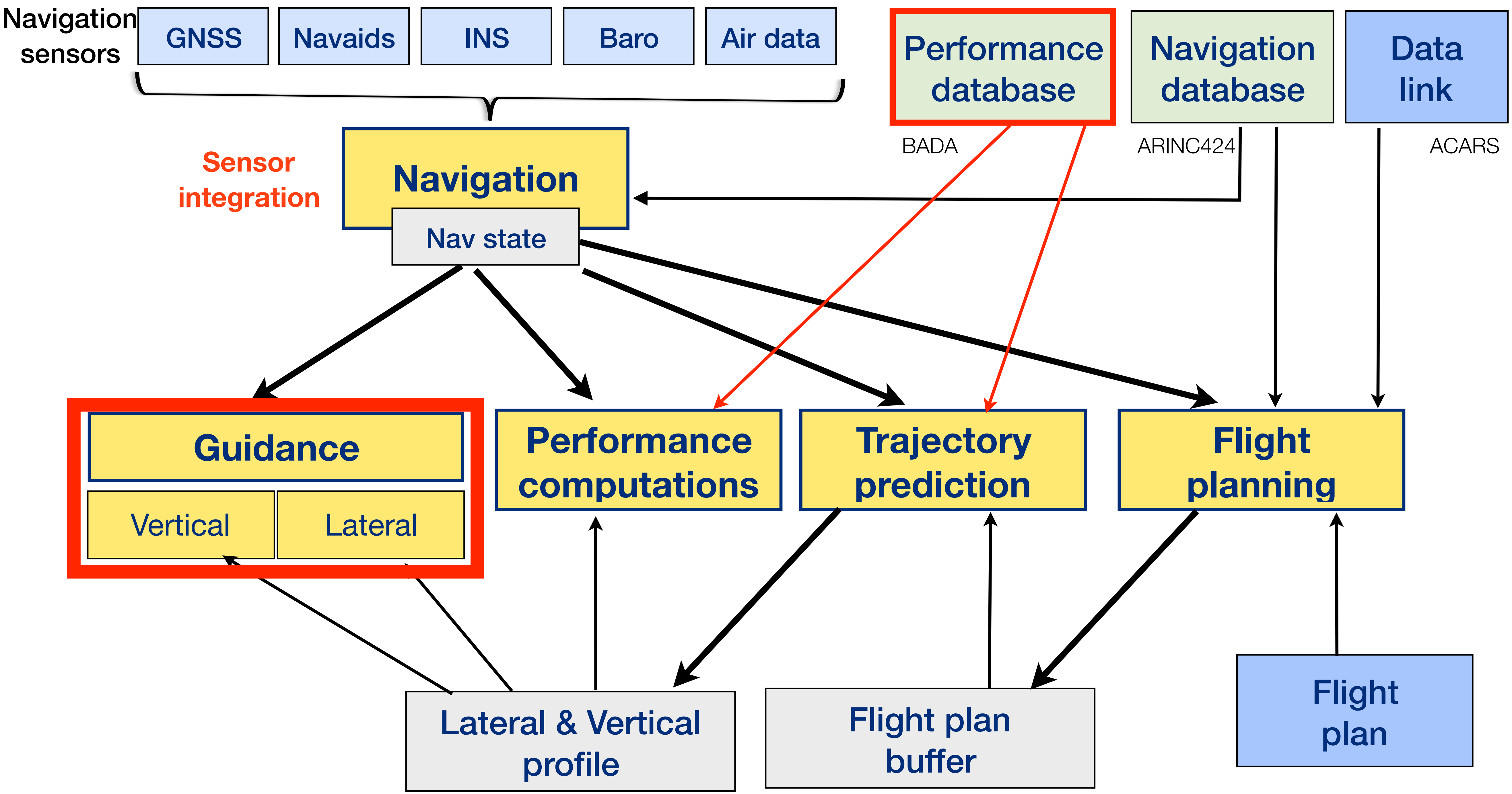
## FMS basic functions

- **Navigation**
  - responsible for determining the best estimate of the current navigation state (position, attitude, ...).
- **Flight planning**
  - allows the crew to establish a specific routing.
- **Trajectory prediction**
  - responsible for computing the predicted aircraft profile along the entire specified routing.
- **Performance computations**
  - provides the crew with aircraft performance information such as takeoff speeds, altitude capability, fuel consumption, optimizations
- **Guidance**
  - producing commands to guide the aircraft along both the lateral and vertical computed profiles



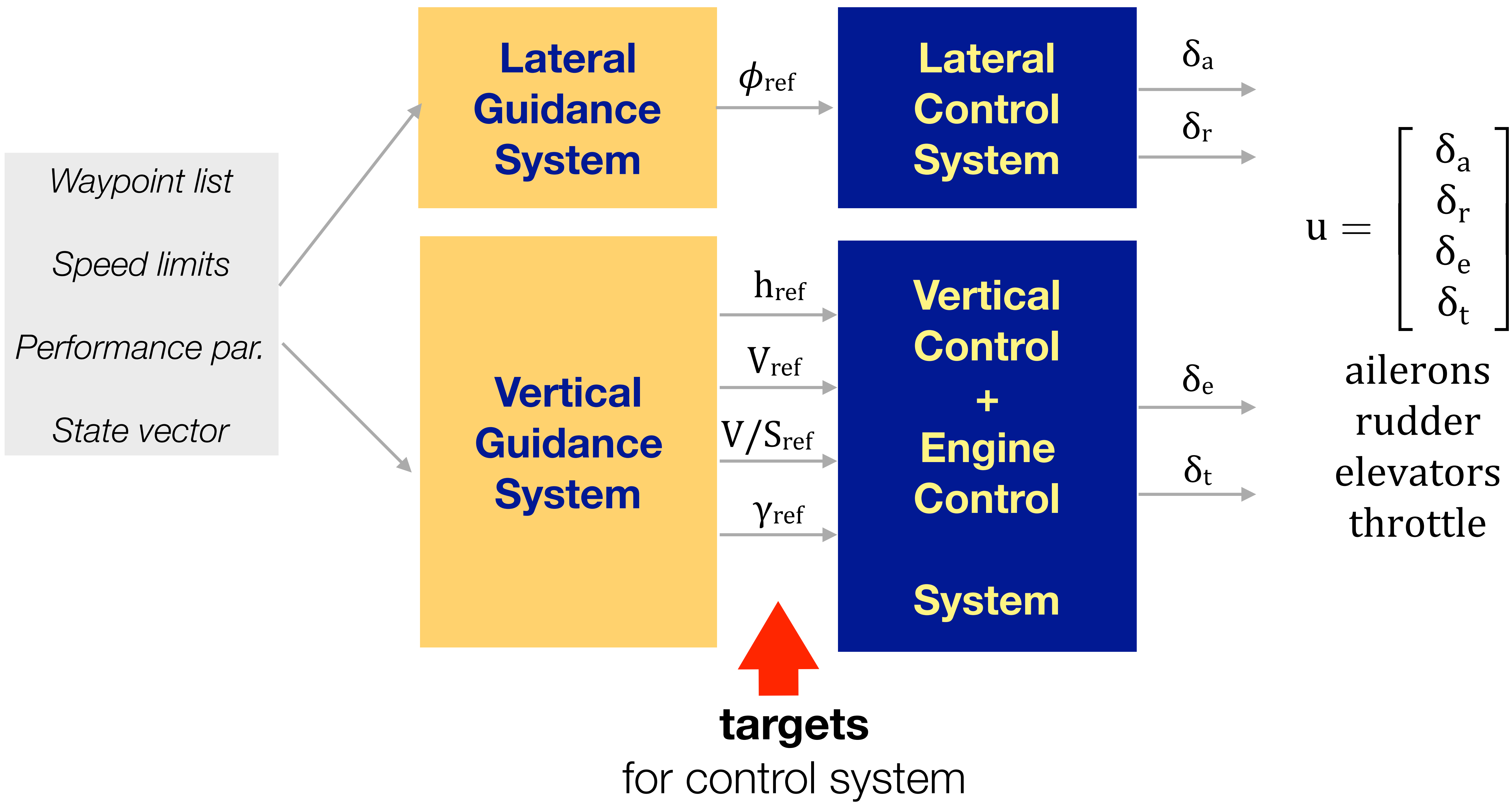
# Introduction

- **FMS functional block diagram**



# Introduction

- General structure of guidance and control systems



# Aircraft guidance

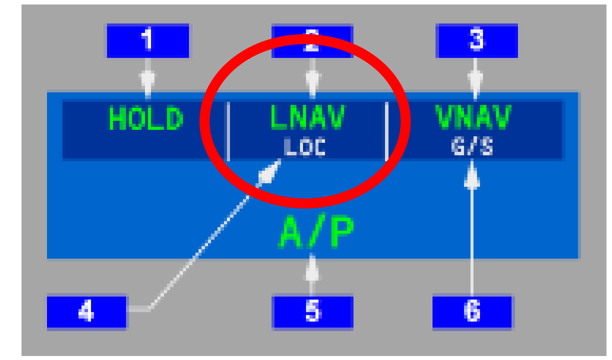


- Introduction
- **Lateral guidance**
- Vertical guidance

# Lateral guidance

## Introduction

- The lateral guidance function provides the different **roll modes** based on the predicted lateral profile described in the trajectory predictions chapter.
- The data are comprised of the classic horizontal situation information:
  - Distance to the active lateral waypoint (DTG)
  - Desired track (DTRK)
  - Track angle error (TRKERR)
  - Cross-track error (XTRK)
  - Drift angle (DA)
  - Bearing to the go to waypoint (BRG)
  - Lateral track change alert (LNAV alert)

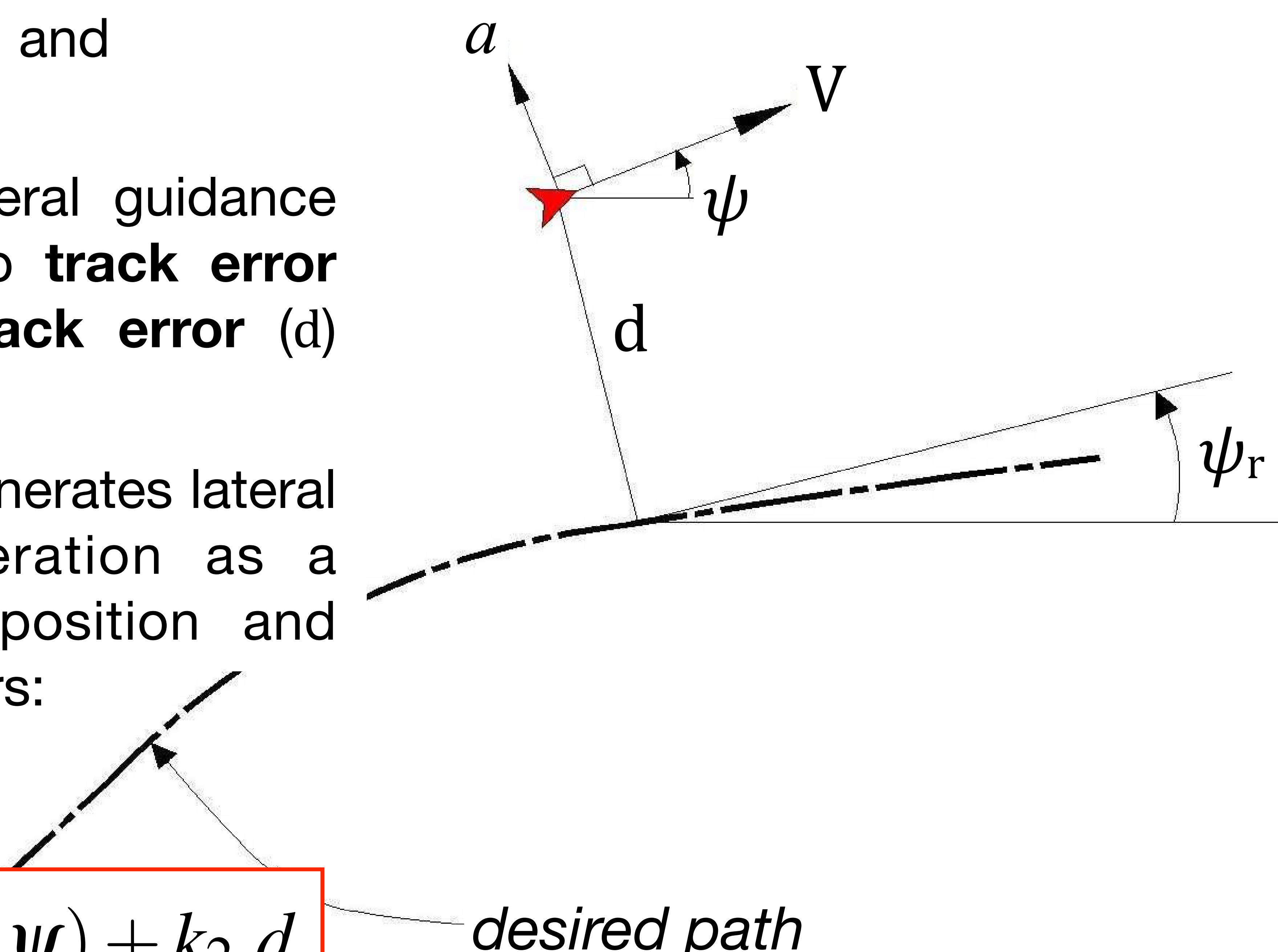



| Roll modes |
|---|
| LNAV (armed) |
| LNAV (engaged) |
| HDG SEL (engaged) |
| TRK SEL (engaged) |
| TRK HOLD (engaged) |
| ATT (engaged) |
| LOC (armed) |
| LOC (engaged) |
| FAC (armed) |
| FAC (engaged) |
| B/CRS (armed) |
| B/CRS (engaged) |
| TO/GA |
| ROLLOUT (armed) |
| ROLLOUT (engaged) |

# Lateral guidance

## Simple lateral guidance

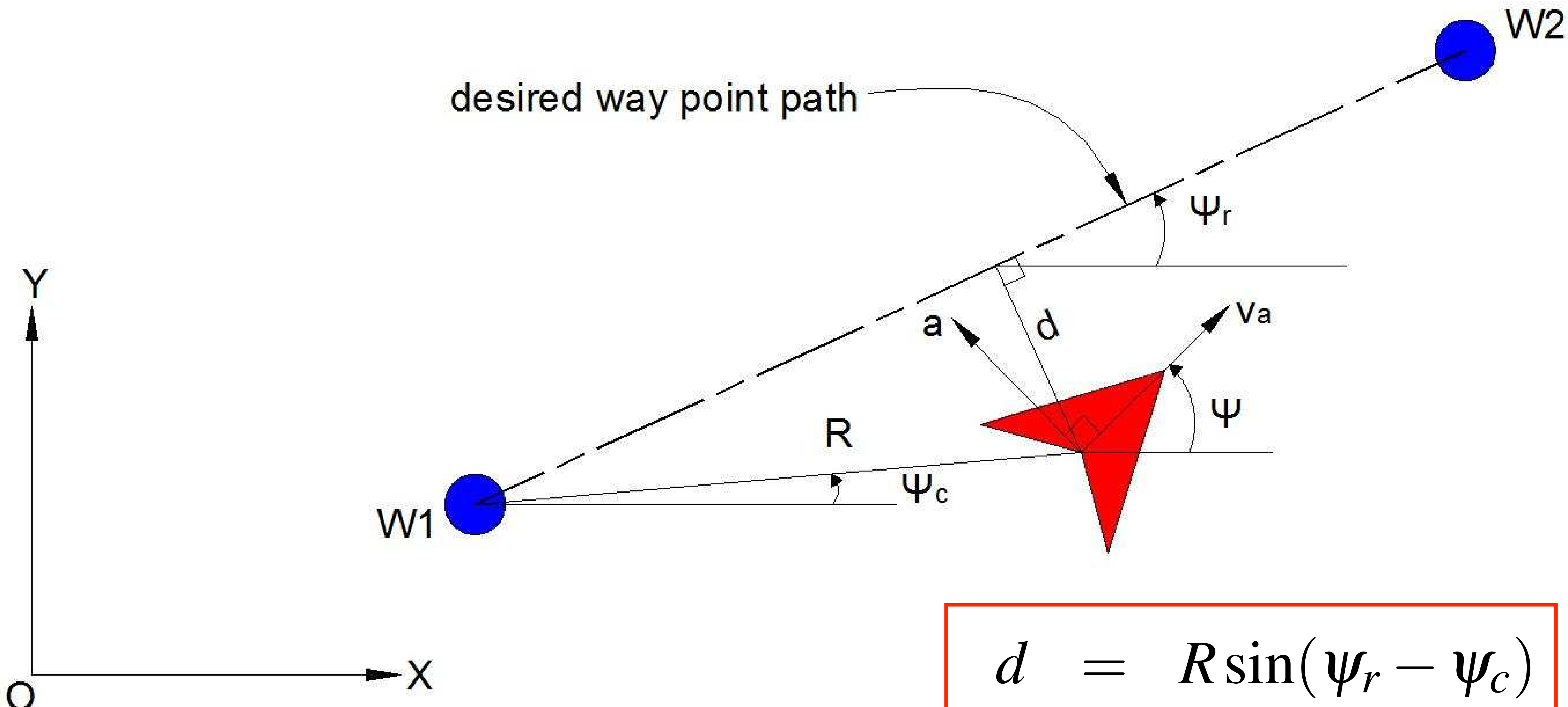
- See *[Walter, R. 2001]* and *[Khotari, M. 2010]*
- The goal of the lateral guidance algorithm is to keep **track error** ($\psi_r$-$\psi$) and **cross-track error** (d) close to zero.
- The guidance law generates lateral (centripetal) acceleration as a weighted sum of position and flight path angle errors:




$$a = k_1(\psi_r - \psi) + k_2\, d$$

# Lateral guidance

- **Simple lateral guidance**
  - Straight-line guidance



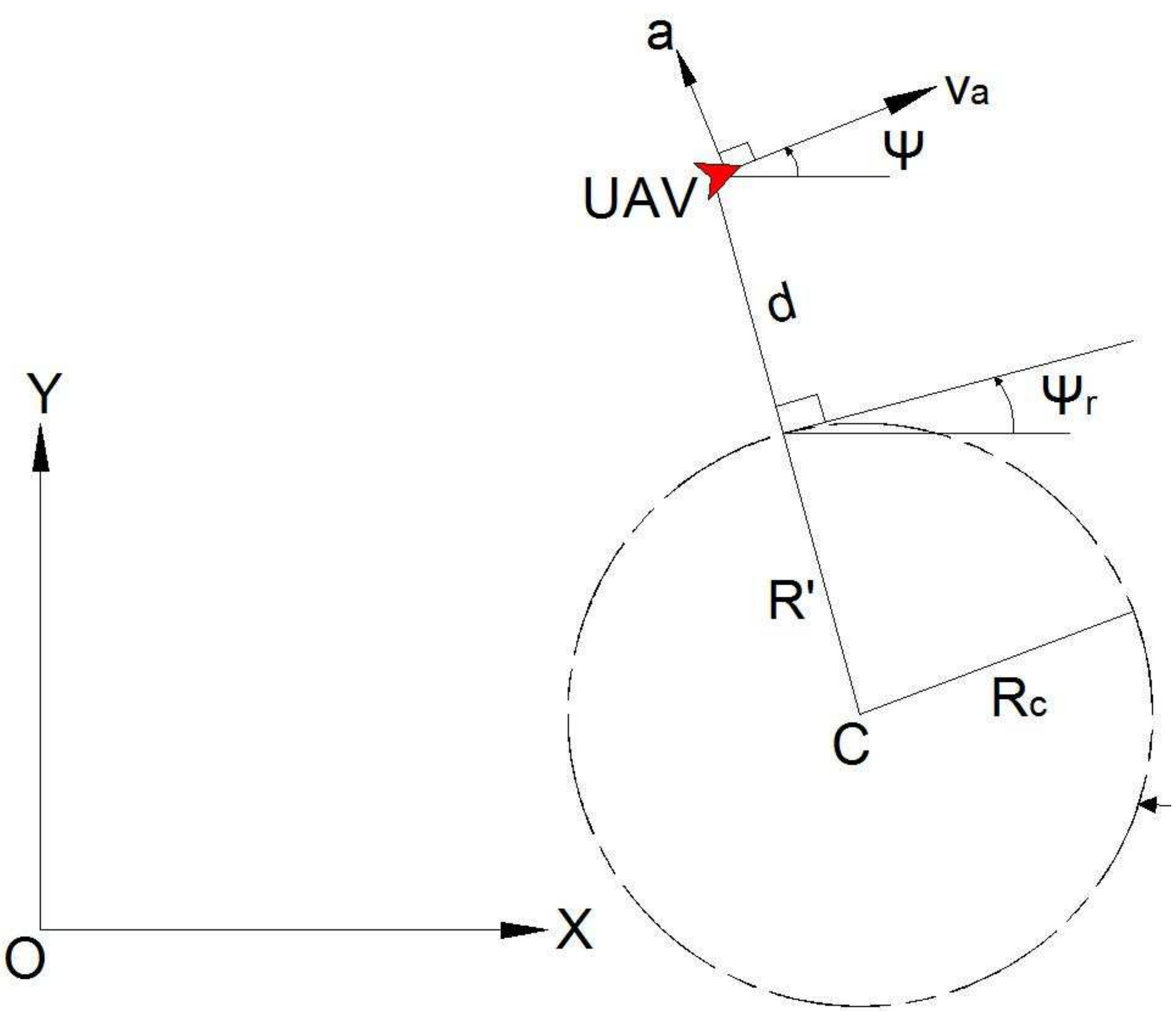

$$
\begin{aligned}
d &= R\sin(\psi_r - \psi_c) \\
\tilde{\psi} &= \psi_r - \psi
\end{aligned}
$$

# Lateral guidance

Flight Management Systems

## Simple lateral guidance

- Circular path following



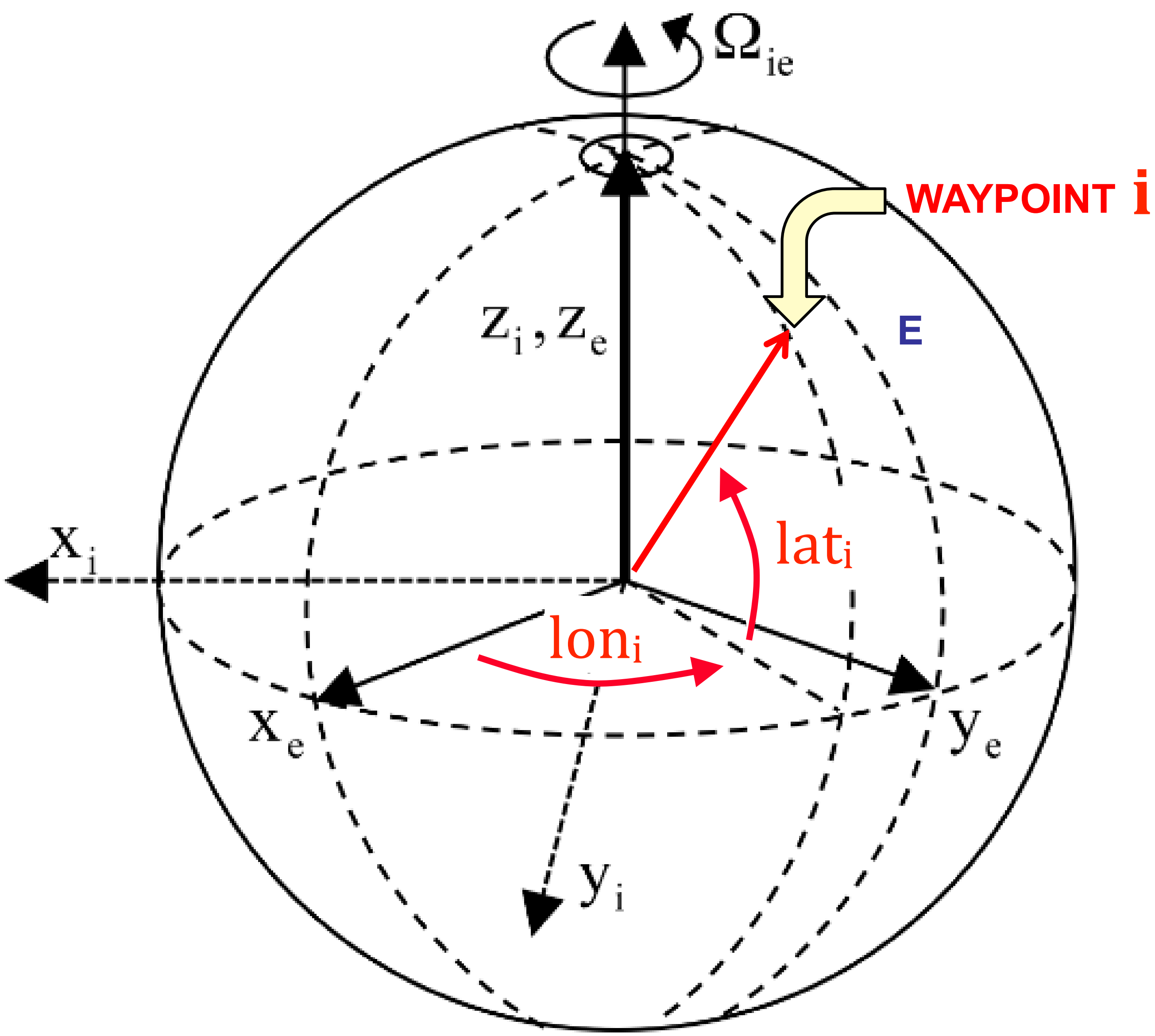

$$\boxed{\begin{aligned} d &= R' - R_c, \\ \tilde{\psi} &= \psi_r - \psi, \end{aligned}}$$

$$R' = \sqrt{(x - xc)^2 + (y - yc)^2)}$$

$$\psi_r = arctan(yc - y, xc - x) \pm arctan(R_c, \sqrt{R'^2 - R_c^2})$$

# Lateral guidance

- **In spherical coordinates**




$$\mathbf{r_i} = R_e \begin{bmatrix} cos(lon_i)cos(lat_i) \\ sin(lon_i)cos(lat_i) \\ sin(lat_i) \end{bmatrix}$$

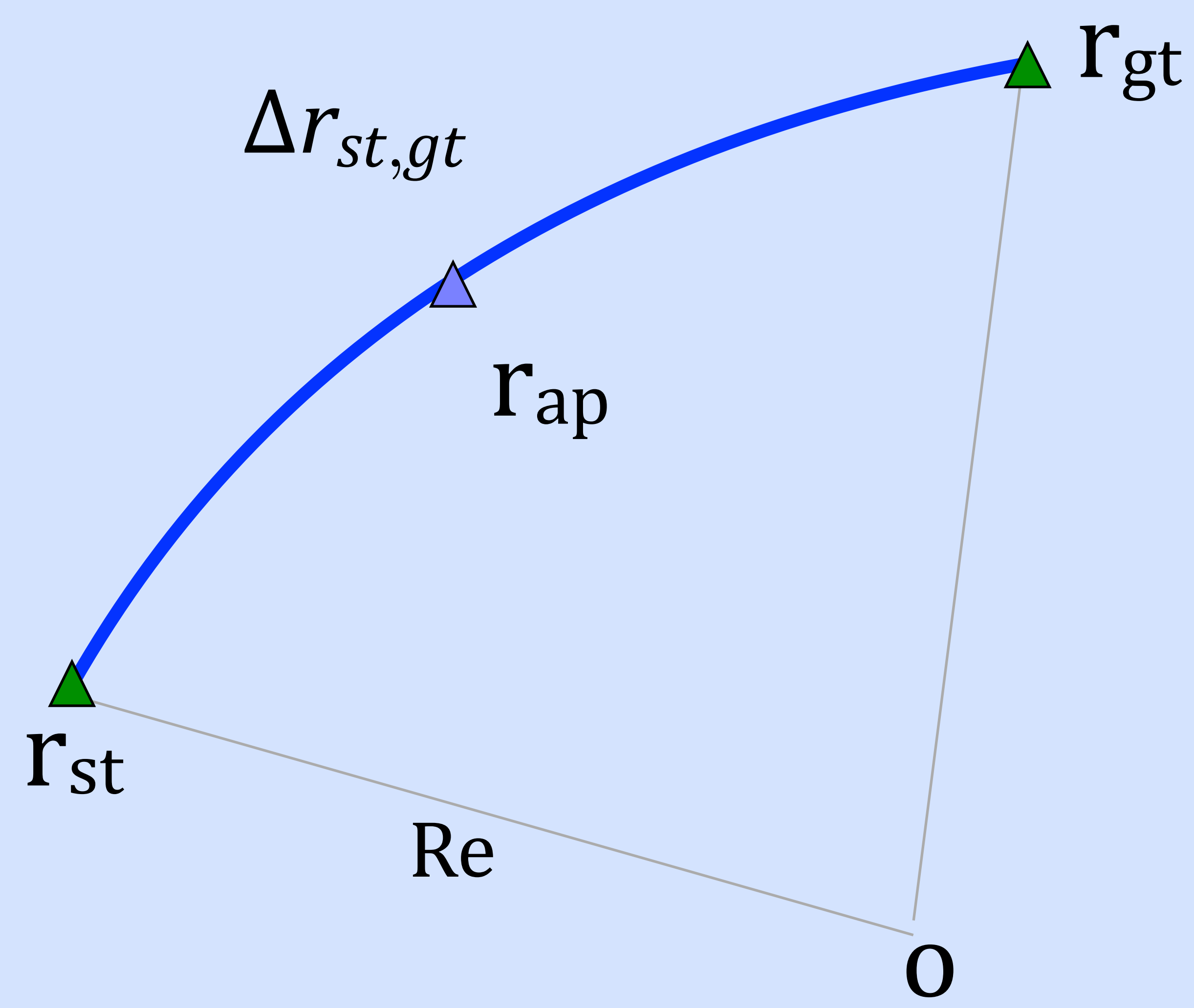
$$\mathbf{r_i} = R_e \mathbf{u_i}$$

Re: Radius of earth

$\mathbf{u_i}$ : Unitary vector

# Lateral guidance

- **In spherical coordinates**




**st** : start point
**gt** : go to
**ap**: along path

$$\Delta r_{st,gt} = R_e cos^{-1}(\mathbf{u}_{st} \cdot \mathbf{u}_{gt})$$

Normal vector to $\mathbf{r}_{st}$ - $\mathbf{o}$ - $\mathbf{r}_{gt}$ plane:

$$\mathbf{n} = \mathbf{u}_{st} \times \mathbf{u}_{gt}$$

East-pointing and Nort-pointing

unit vectors at $\mathbf{r}_{st}$

$$\mathbf{u}_{st}^{E} = \mathbf{u}_{Z} \times \mathbf{u}_{st}$$
$$\mathbf{u}_{st}^{N} = \mathbf{u}_{st} \times \mathbf{u}_{st}^{E}$$

where $\mathbf{u}_{z}$ is Z axis unit vector

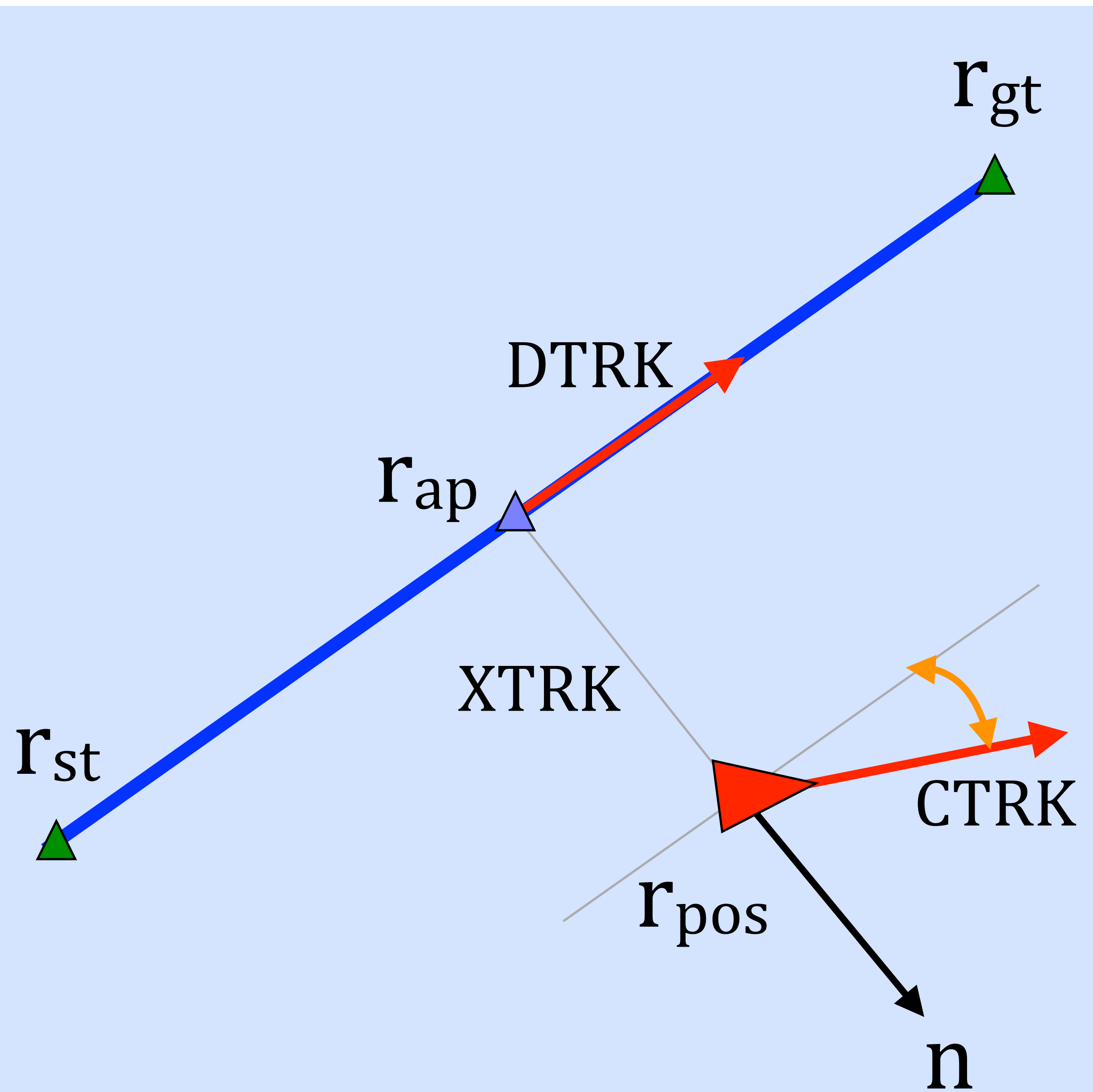
$$\mathbf{u}_{Z} = [0, 0, 1]^{T}$$

# Lateral guidance



- In spherical coordinates

**Top view**




**Desired track**:

$$DTRK = atan\left(\frac{-\mathbf{n}\cdot\mathbf{u}_{ap}^{N}}{-\mathbf{n}\cdot\mathbf{u}_{ap}^{E}}\right)$$

**Track error:**

Desired TRK - Current TRK

$$TRKERR = DTRK - CTRK$$

**Cross-track error:**

*(full expression)*

$$XTRK = -R_e cos^{-1}(\mathbf{u}_{ap}\cdot\mathbf{u}_{pos})$$

*(good approximation)*

$$XTRK = -R_e \mathbf{n}\cdot\mathbf{u}_{pos}$$

# Lateral guidance

- **Guidance law for flyby waypoints**
  - **Capture**:

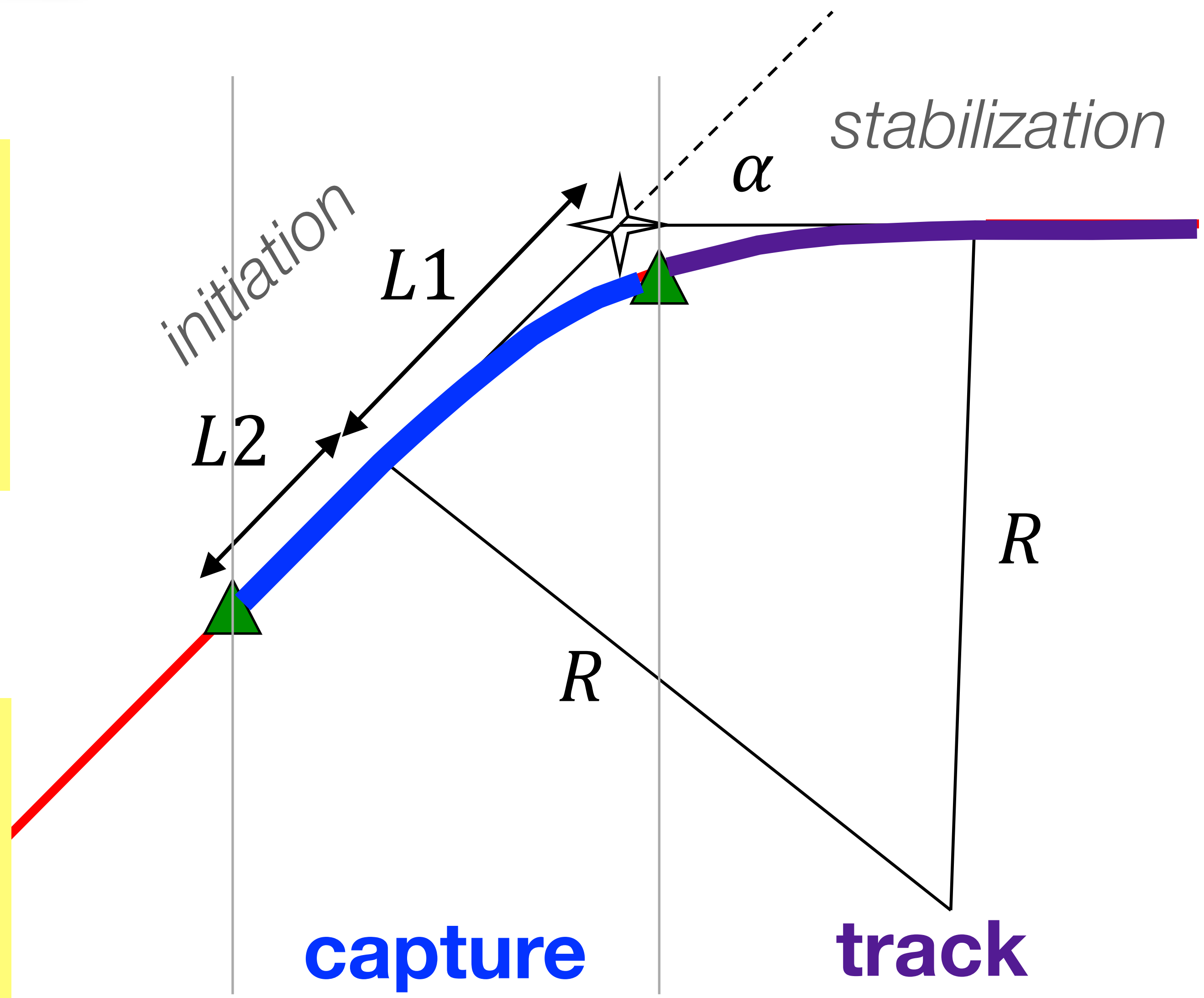
$$\varphi_{nominal} = tan^{-1}\frac{V \cdot \dot{\psi}}{g}$$

  - **Track**:

$$\varphi_{commanded} = K_{XTRK} \cdot XTRK + K_{XTRERR} \cdot TRKERR$$

  - **Transition** from capture to track phase when:

$$\varphi_{commanded} < \varphi_{nominal}$$



# Lateral guidance



## Flyby transitions

- Anticipation of the activation of the next vertical leg is required, to accomplish a smooth capture of that segment is performed without path overshoot.
- Turn initiation criteria

**L2** selected based on how quickly the aircraft responds to a change in the aileron position.

$$R = \frac{V^2}{g \cdot tan\varphi_{nominal}} \qquad L = \underbrace{R \cdot tan(0.5\alpha\ )}_{L1} + \underbrace{roll\ in\ distance}_{L2}$$

$$L2 = cV$$

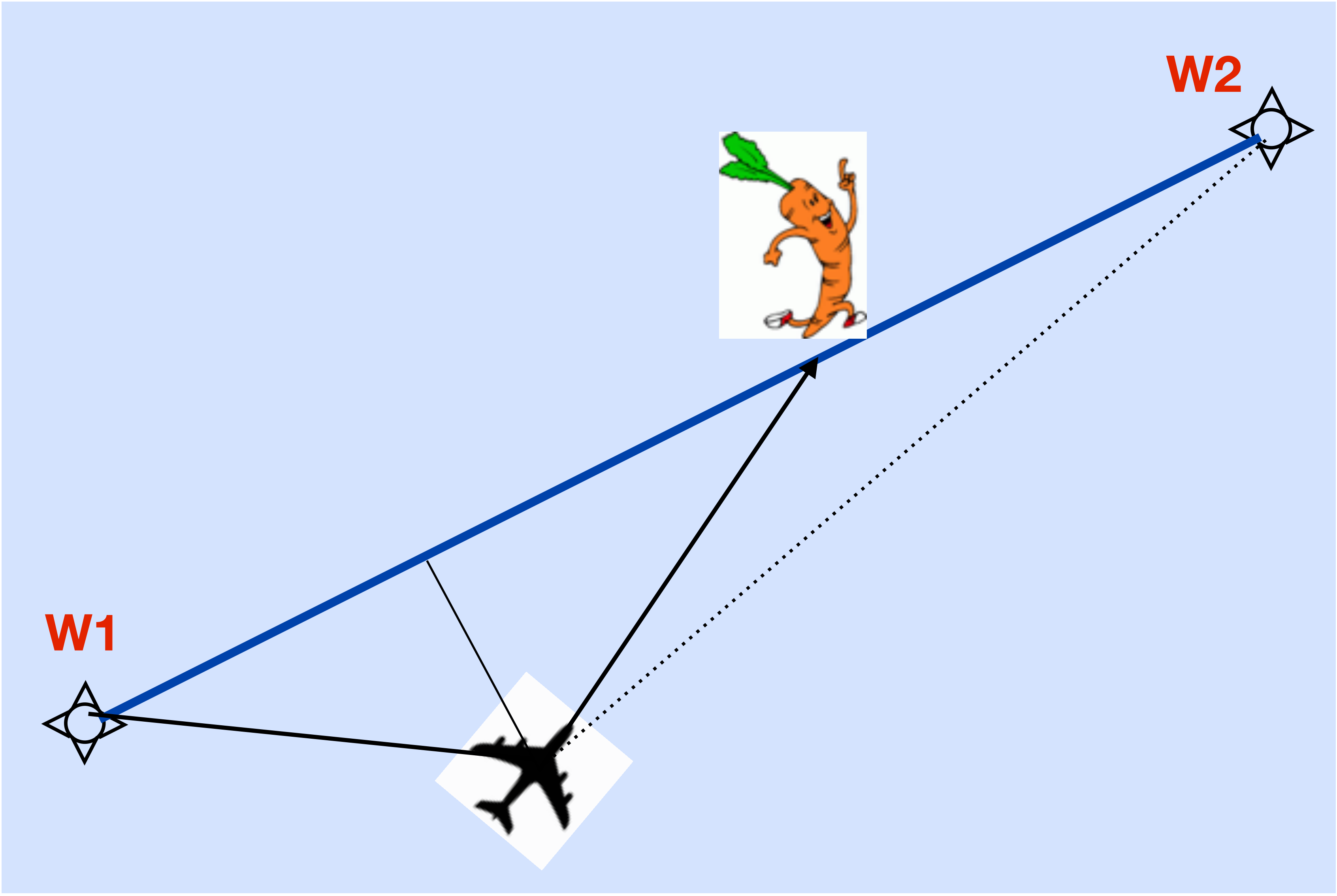
## Flyover transitions

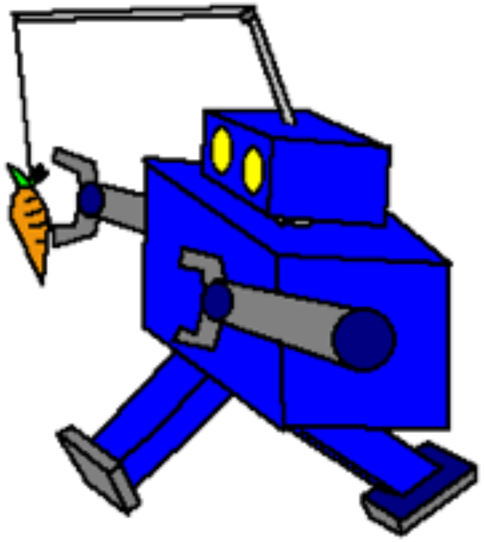
- The activation of the next leg occurs at the time the “flyover” waypoint is sequenced.

# Lateral guidance



- **The "carrot algorithm"**
  - See *[Ducard, GJJ. 2009]*
  - The aircraft tries to follow a reference point (*"carrot"*) that moves along the desired path

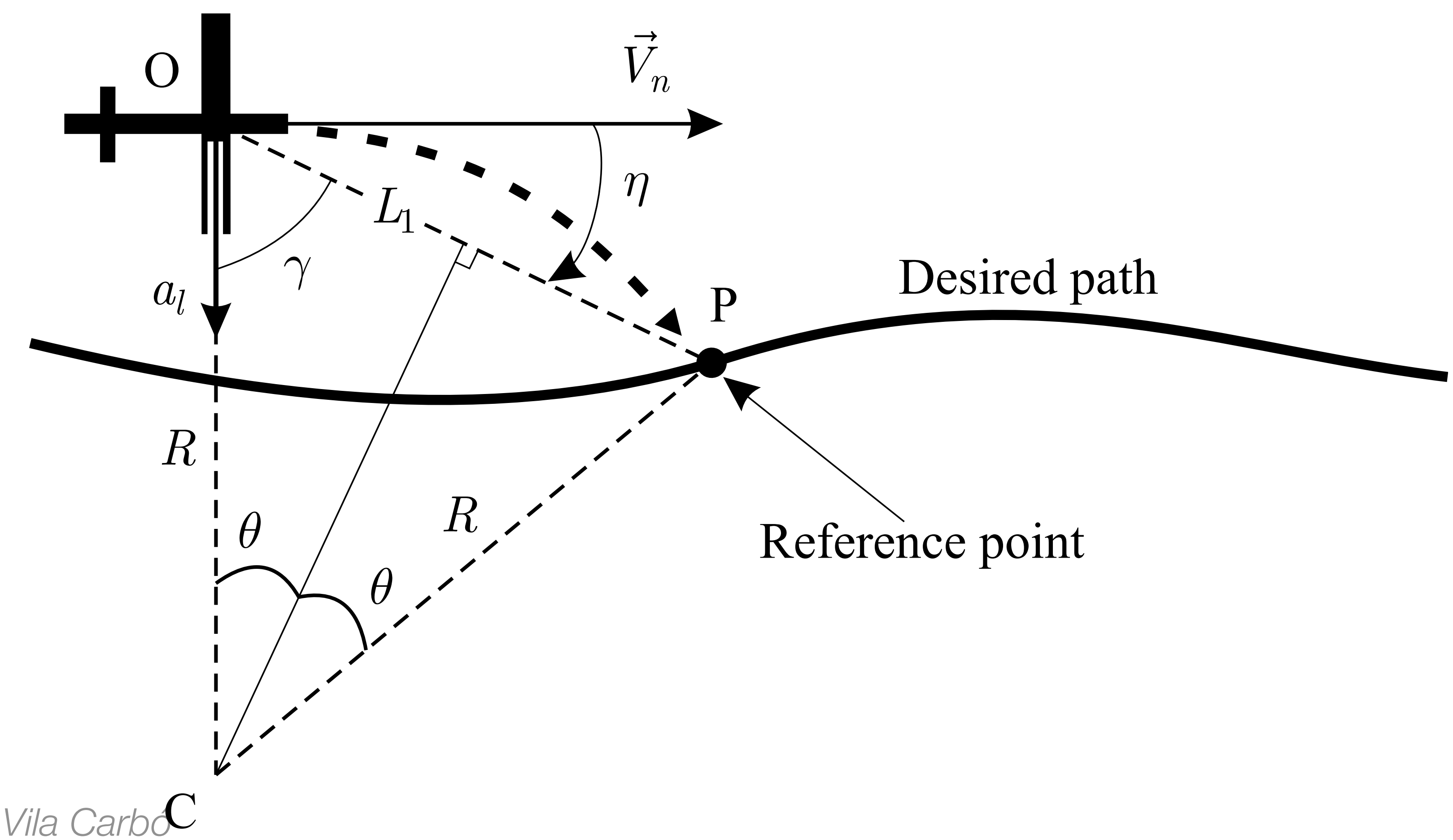


# Lateral guidance

## The “carrot algorithm”

- **Setting the reference point**: from the current location O, draw an arc of radius R that intersects the desired path at a “reference point” P.
  - *L1* is the segment that joins the aircraft O and the reference point P
  - η is the angle between the aircraft’s velocity vector and the line *L1*.
- **Lateral acceleration** required to bring the aircraft to the reference point:




$$a_l = \frac{V_n^2}{R} \,,$$

$$L_1 = 2R\sin\eta \,,$$

$$\boxed{a_l = \frac{2V_n^2}{L_1}\sin\eta}$$

## Lateral guidance

### The "carrot algorithm"

- Computation of the Roll Angle command
  - The direction of the aircraft's ground speed is computed with:

$$\psi = \tan^{-1}\left(\frac{V_E}{V_N}\right) \quad \in [-\pi; \pi]$$

  - The lateral guidance control law needs the angle η, which is computed as:

$$\eta = \Omega - \psi \ ,$$
$$\Omega = \tan^{-1}\left(\frac{P_E - X_E}{P_N - X_N}\right) \quad \in [-\pi; \pi]$$

$$\text{while } (\eta > \pi) \ \ \eta = \eta - 2\pi \ ,$$
$$\text{while } (\eta < -\pi) \ \ \eta = \eta + 2\pi \ .$$

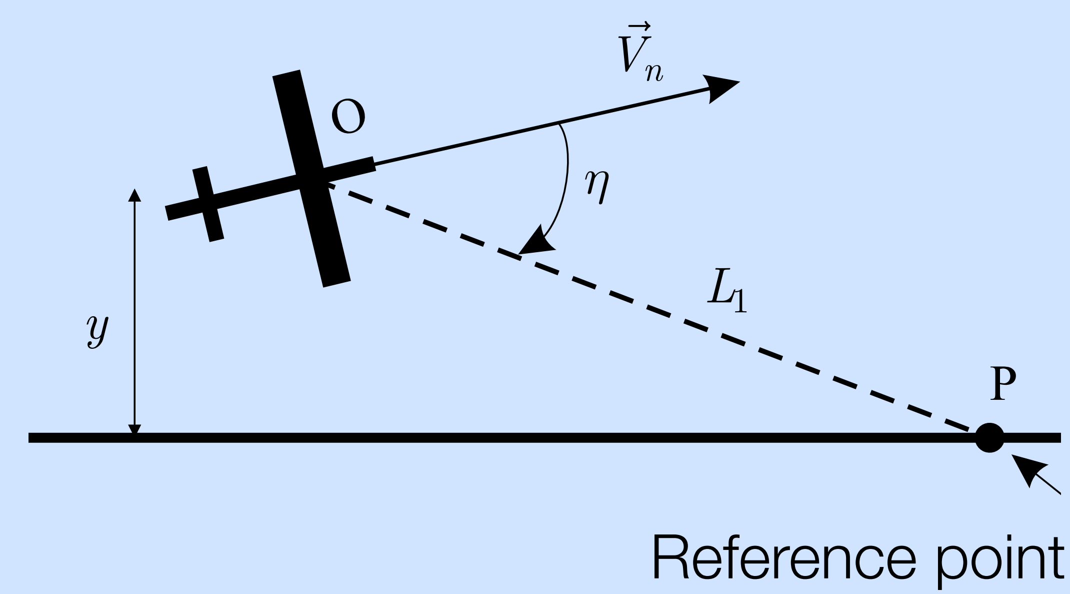
$$\varphi_{commanded} = \frac{2V_n^2 sin(\eta)}{L_1 g}$$

# Lateral guidance

## The "carrot algorithm"

- For a small magnitude of η, the guidance formula can be approximated in terms of the cross-track error y as follows:

$$a_l = \frac{2V_n^2}{L_1}\sin\eta \approx 2\frac{V_n}{L_1}\left(\dot{y} + \frac{V_n}{L_1}y\right)$$




- The guidance law behaves as a PD controller, in which the ratio $V_n / L_1$ is the gain of the controller.

- The control gain is limited by the inner loop bank control bandwidth (2-3 rad/s).
- With a nominal flight velocity of around 25 m/s, the distance L1 has been chosen to be L1 = 150 m

# Lateral guidance

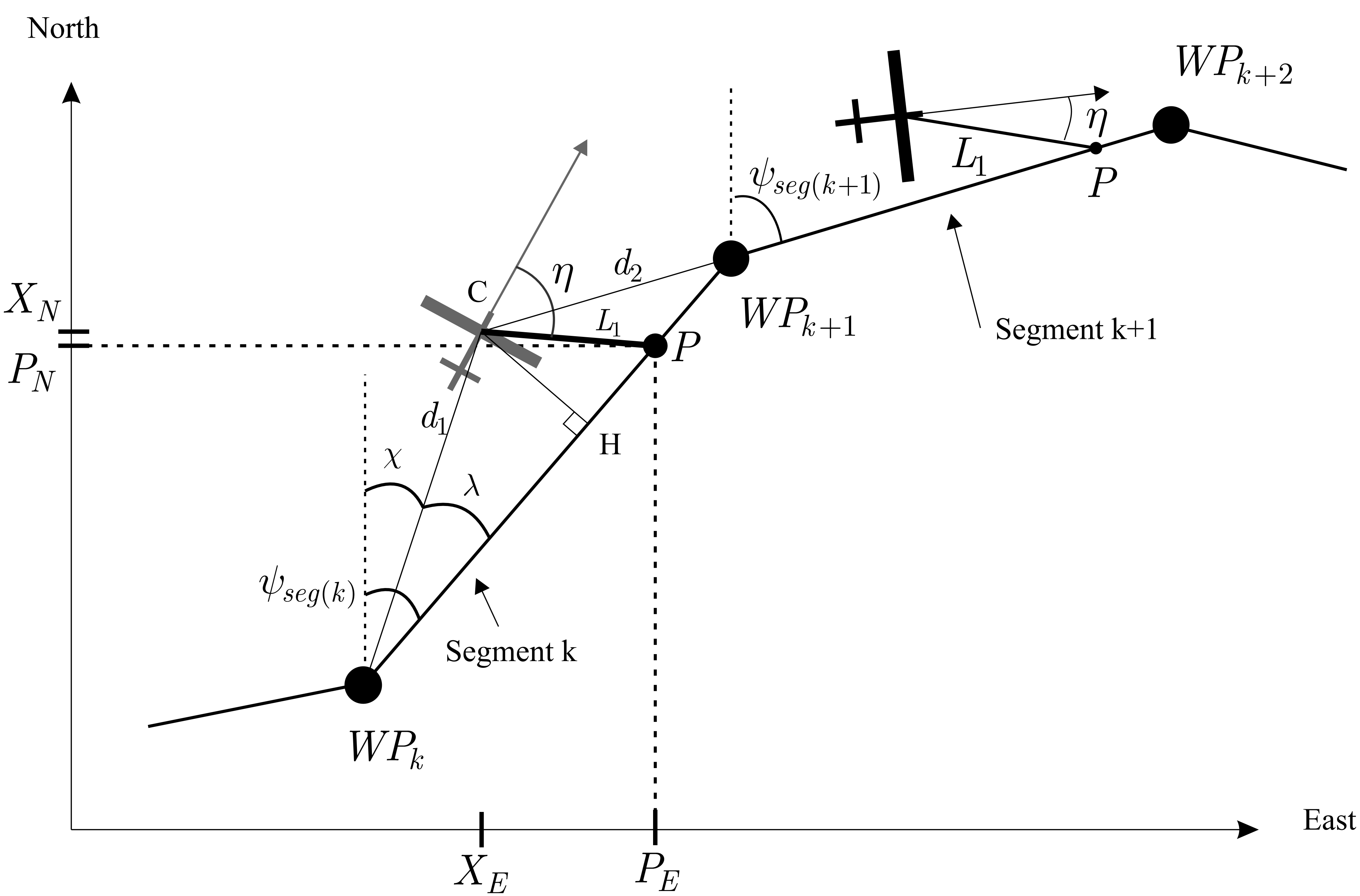
Flight Management Systems

## The “carrot algorithm”

- This control law is particularly suited to fly circles.
  - If the aircraft is following a desired circular path, then the acceleration command $a_l$ generated by the guidance system is exactly the same as the associated centripetal acceleration.
  - In other words, **the guidance logic** chooses a reference point on the desired path at a distance L1 ahead of the aircraft, and **generates the acceleration command that would lead the vehicle to hit the point after flying a circular arc**, thus leading to zero steady-state error for a circular path.

# Lateral guidance

- **The "carrot algorithm"**
  - Regular waypoint tracking



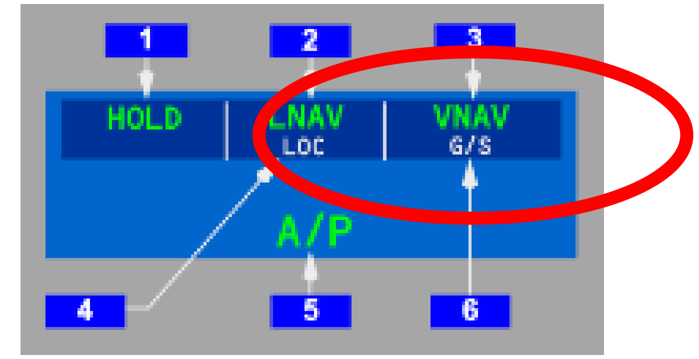
# Aircraft guidance

- **Introduction**
- **Lateral guidance**
- **Vertical guidance**

# Vertical guidance

## Introduction

- Provides commands of **pitch** and **thrust control** to accomplish target speeds, target thrusts, target altitudes, and target vertical speeds.
- Unlike the lateral guidance parameters, the vertical guidance parameters are somewhat flight phase dependent.




| Autothrottle modes | Pitch modes |
|---|---|
| THR | TO/GA |
| THR REF | VNAV (armed) |
| HOLD | VNAV SPD (engaged) |
| IDLE | VNAV PTH (engaged) |
| SPD | VNAV ALT (engaged) |
| | V/S (engaged) |
| | FPA (engaged) |
| | FLCH SPD (engaged) |
| | ALT (engaged) |
| | G/S engaged) |
| | G/P (engaged) |
| | FLARE (armed) |
| | FLARE (engaged) |

# Vertical guidance

## Vertical guidance - data

| Flight Phase | Vertical Guidance Data |
|---|---|
| Takeoff | Take-off speeds V1, V2, VR<br>Take-off thrust limit |
| Climb | Target speed based on selected climb speed schedule, flight plan speed restriction, and airframe limitations<br>Target altitude intercept<br>Alt constraint violation message<br>Distance to top of climb<br>Climb thrust limits |
| Cruise | Target speed based on selected cruise speed schedule, flight plan speed restriction, and airframe limitations<br>Maximum and optimum altitude distance to step climb point<br>Distance to top of descent<br>Cruise thrust limit<br>Cruise thrust target |

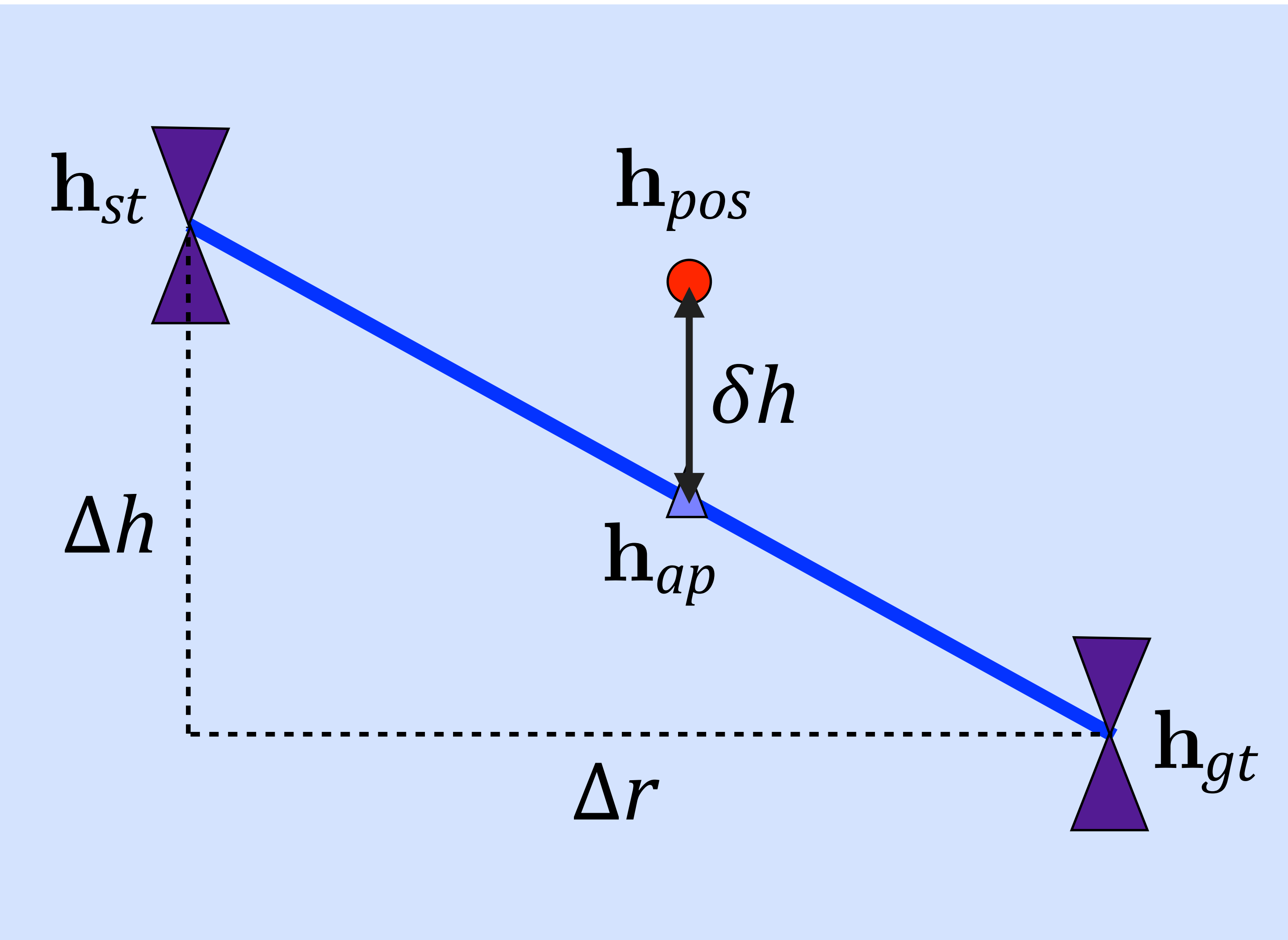
# Vertical guidance

## Vertical guidance - data (ii)

| Flight Phase | Vertical Guidance Data |
|---|---|
| Descent | Target speed based on selected descent speed schedule, flight plan speed restriction, and airframe limitations |
| | Target altitude intercept |
| | Vertical deviation |
| | Desired V/S |
| | Energy bleed-off message |
| Approach | Target speed based on dynamic flap configuration |
| | Vertical deviation |
| | Desired V/S |
| Missed Approach | Target speed based on selected climb speed schedule, flight plan speed restriction, and airframe limitations |
| | Target altitude intercept |
| | Alt constraint violation msg |
| | Distance to top of climb |
| | Go-around thrust limit |

# Vertical guidance

- **Interpolating data**




Vertical deviation:

$$\delta h = \mathbf{h}_{pos} - \mathbf{h}_{ap}$$

Path gradient:

$$g_h = \frac{\delta h}{\delta r}$$

**Path altitude:**

$$\mathbf{h}_{ap} = \mathbf{h}_{gt} + g_h \Delta r_{ap-gt}$$

**Desired $V/S$:**

$$V/S_{desired} = \dot{h}_{desired} = g_h V_{GS}$$

# Vertical guidance



## Computing speed and thrust

- The **target speed** data are usually *not interpolated* from the predicted vertical profile since it is only valid for on-path flight conditions.
- Instead, it is *predicted* based on the current flight phase, aircraft altitude, relative position with respect to flight plan speed restrictions, flaps configuration, and airframe speed envelope limitations.
- This applies to **thrust limit** computations as well.

# Vertical guidance

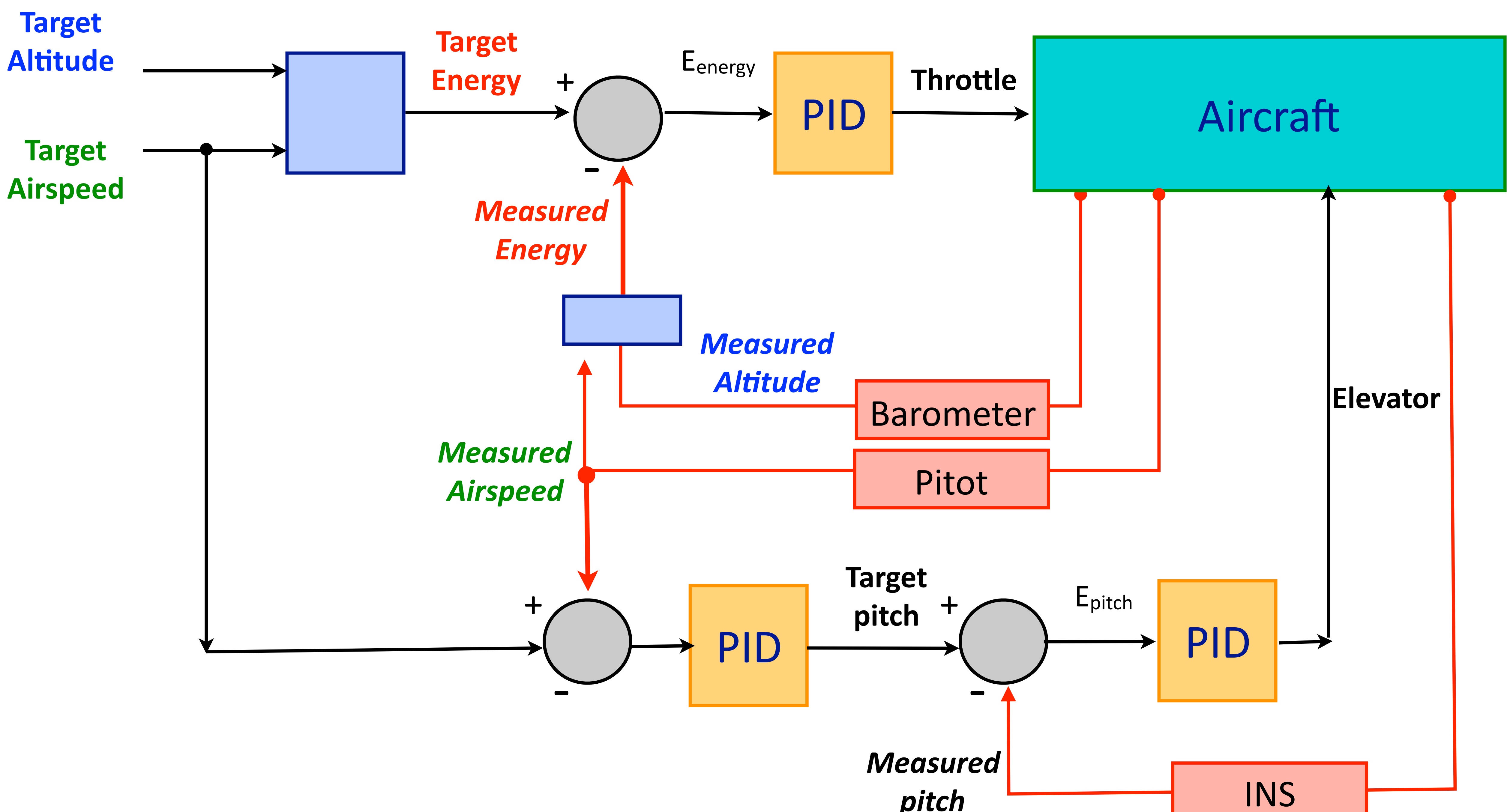
## Pitch and thrust axis setting

- Based on tracking the **speed** target, FMS **path**, or acquiring and **holding** a target altitude depending on the flight phase and situation.
  - An annunciation of the parameter controlling pitch is displayed in the FMC

| Flight Phase | Pitch Axis Control | Thrust Axis Control | Pitch/Thrust Mode Annunciation |
|---|---|---|---|
| Take-off | None until safely off ground then same as climb | Take-off thrust limit | Vspd/TO limit |
| Climb and cruise climb | Capture and track speed target | Climb thrust limit | Vspd/CLB limit |
| Level flight | Capture and maintain altitude | Maintain speed target | Valt/CRZ limit |
| Unrestricted descent | Capture and track vertical path | Set to flight idle | Vpath/CRZ limit |
| Restricted descent and approach | Capture and track vertical path | Set to computed thrust required, then maintain speed | Vpath/CRZ limit |
| Descent path capture from below and cruise descent | Capture and track fixed V/S capture path | Set to computed thrust required, then maintain speed | Vpath/CRZ limit |
| Descent path capture from above | Capture and track upper speed limit | Set to flight idle | Vspd/CRZ limit |
| Missed approach | Capture and track speed target | Go-around thrust limit | Vspd/GA limit |

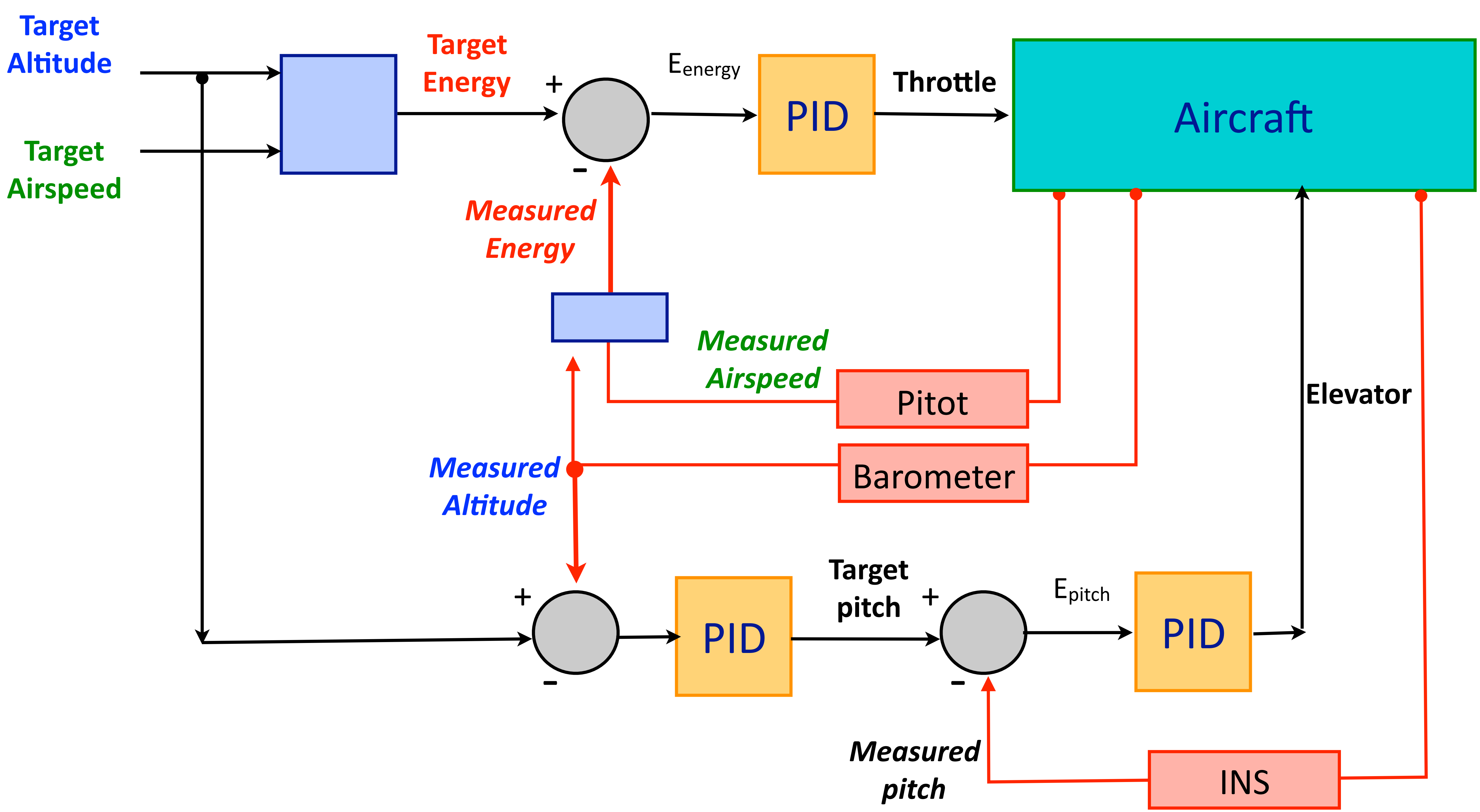
# Vertical guidance

- Control loop for **Vspd**



# Vertical guidance

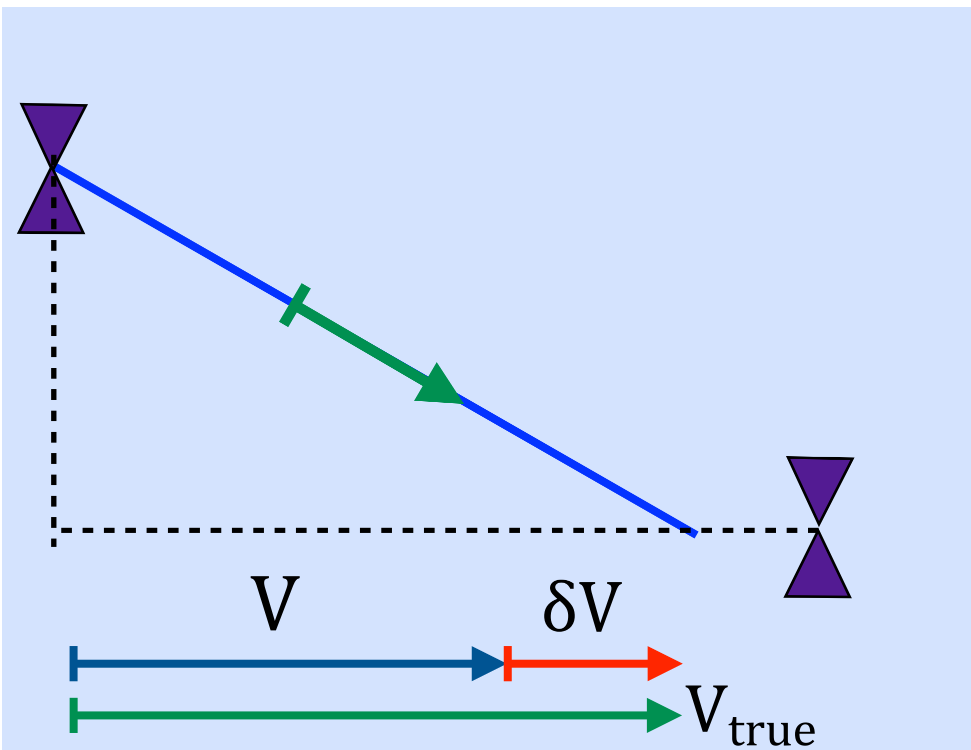
- Control loop for **Vpath**



# Vertical guidance

## Pitch axis control

- **Vspd**: constant speed




**Capture:**

$$\Delta\theta = K_{speed-rate} * (\dot{V} - \dot{V}_{capture})$$

**Track:**

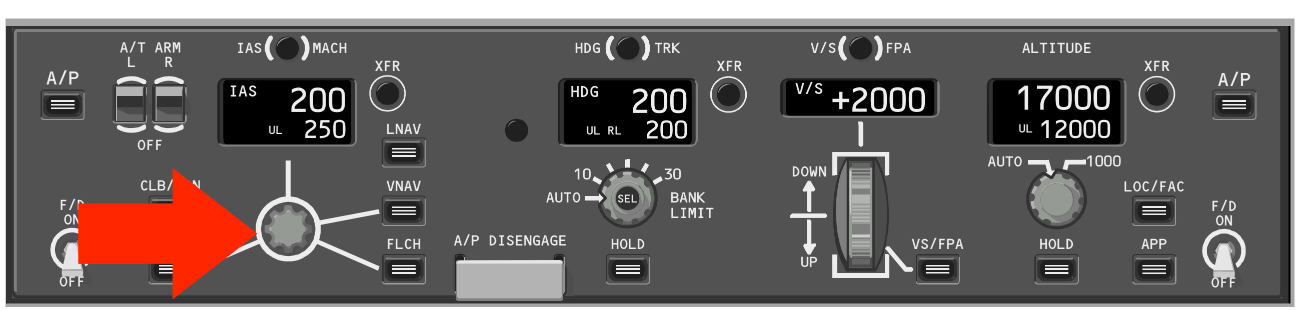
$$\Delta\theta = (K_{airspeed} * \delta V + K_{speed-rate}\dot{V})/V_{true}$$

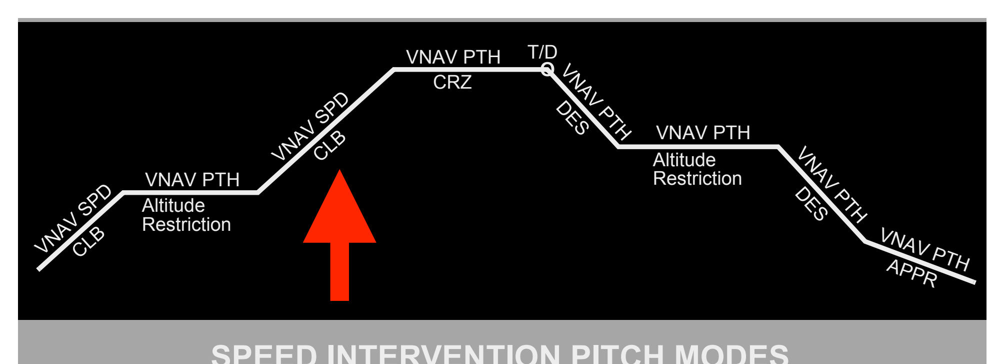
Airspeed error

Airspeed rate

For example VNAV SPD (B787 FCOM):

in the **VNAV SPD** pitch mode, the AFDS commands pitch to hold target airspeed (“track”). The autothrottle operates in the THR REF, THR, IDLE or HOLD mode, as required by the phase of flight.



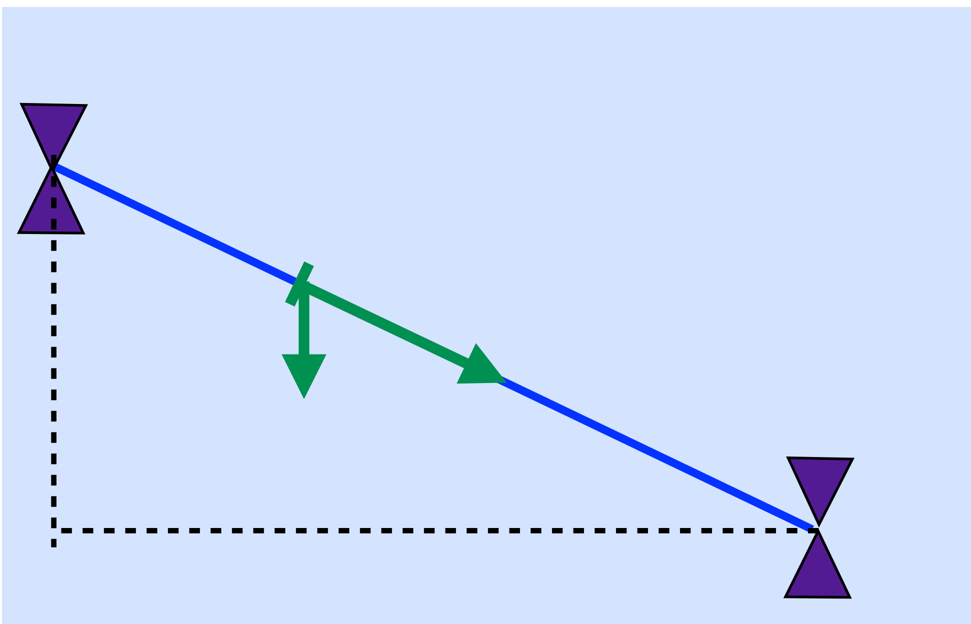



# Vertical guidance

Flight Management Systems

## Pitch axis control

- **Vpath**: geometric path



$$\delta \dot{h} = V/S_{error} = V/S_{fixed\ capture} - V/S_{current}$$

**Capture:**

$$\Delta\theta = K_{capture} * sin^{-1}\left(\frac{\delta \dot{h}}{V_{true}}\right)$$

**Track:**

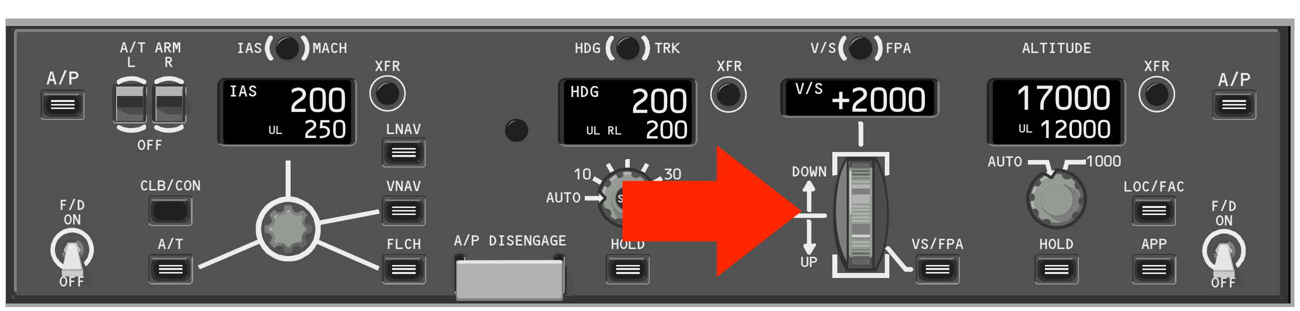
$$\Delta\theta = (K_{ALT} * \delta h + K_{V/S} * \delta \dot{h})/V_{true}$$

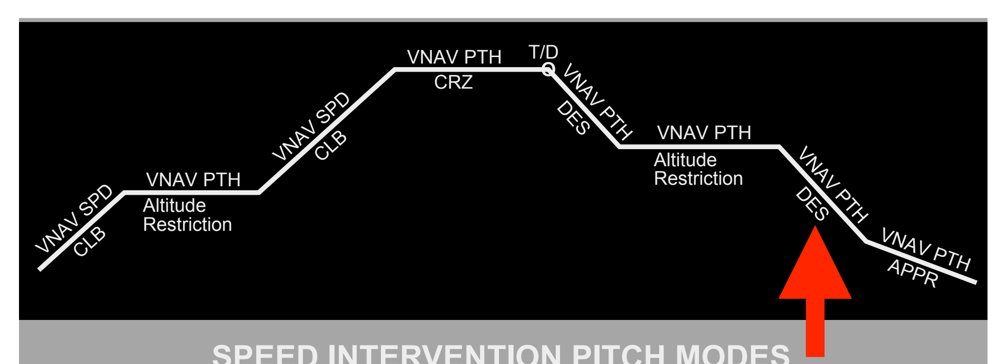
Altitude error

V/S

For example VNAV SPD (B787 FCOM):

in the **VNAV PTH** pitch mode, the AFDS commands pitch to maintain FMC target altitude or the VNAV path. The autothrottle maintains speed.



# Vertical guidance

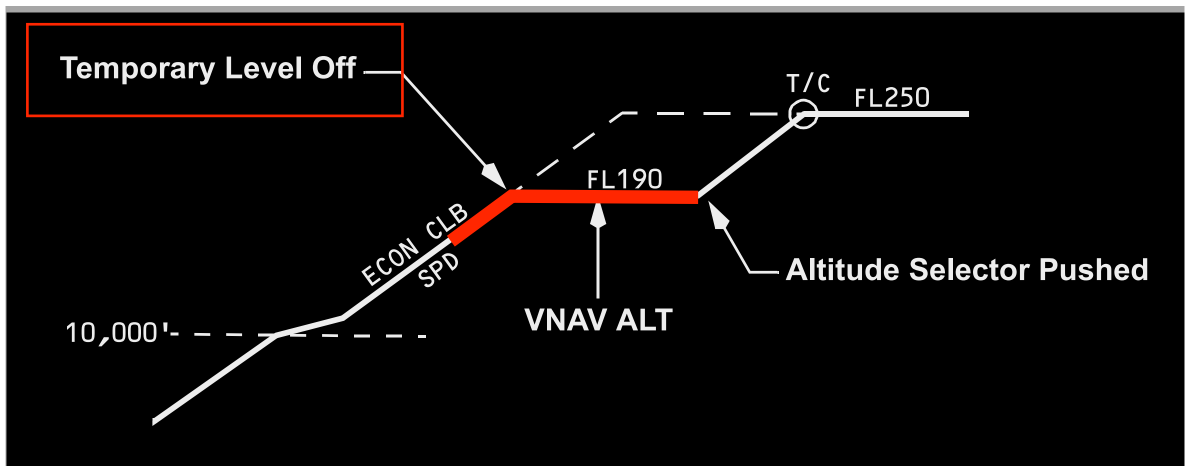
Flight Management Systems

## Pitch axis control

- **Valt**: geometric path

$$V/S_{capture} = K_{alt-capture} * \delta h$$

$$\delta\dot{h} = V/S_{error} = V/S_{capture} - V/S_{current}$$




ALTITUDE INTERVENTION EXAMPLE

**Capture:**

$$\Delta\theta = K_{V/S} * sin^{-1}\left(\frac{\delta\dot{h}}{V_{true}}\right)$$

**Track:**

$$\Delta\theta = (K_{ALT} * \delta h + K_{V/S} * \delta\dot{h})/V_{true}$$

Altitude error V/S

For example VNAV SPD (B787 FCOM):

in the VNAV ALT pitch mode, the AFDS commands pitch to maintain the MCP selected altitude when that altitude is lower than the VNAV commanded altitude in climb or higher than the VNAV commanded altitude in descent .

# Thrust control



## Computing required thrust

- **Thrust setting for maintaining a speed** is only used for an initial throttle setting.
- Thereafter the **speed error is used to control the throttles.**
- **Thrust required** is set based on the equations of motion:

$$T = \frac{W \times V/S_{ave}}{V_{ave}}\left(1 + \frac{V_{ave}}{g}\frac{dV_{true}}{dh}\right) + D$$

- **Thrust limit** and **Idle thrust** must be taken into account (see BADA).

**Thrust Limit**

*Thrust limit = f (temp, alt, spd, engine bleed air): stored as data sets in the performance DB*

**Flight Idle**

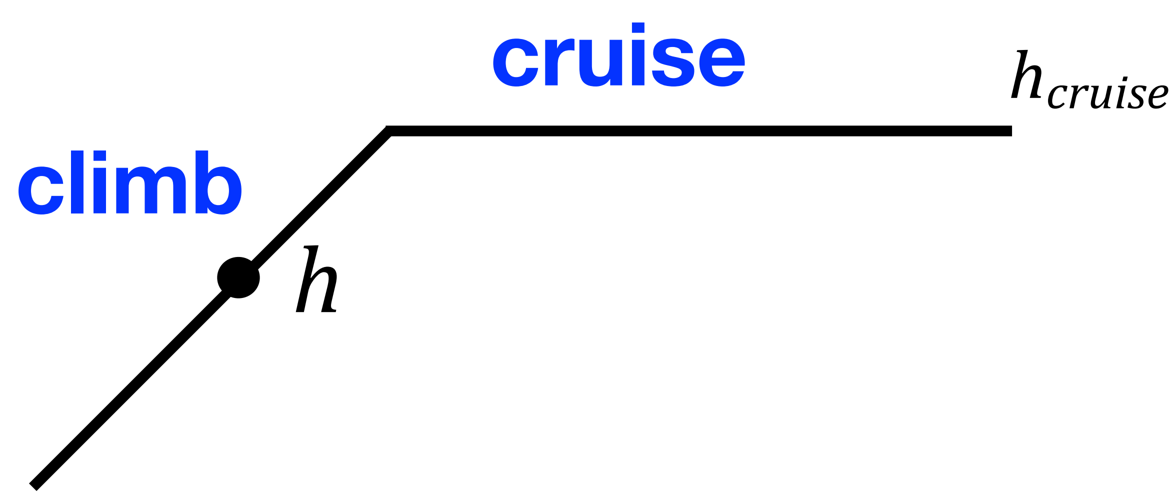
*Idle thrust = f (temp, alt, spd, engine bleed air): stored as data sets in the performance DB*

# Vertical guidance

## Flight phase transitions

- The selected altitude is used as a limiter in that the vertical guidance will not allow the aircraft to fly through that altitude
  - except during approach operations where the selected altitude may be pre-set for a missed approach if required).
- When on the ground with the flight plan and performance parameters initialized, the flight phase is set to take-off. After liftoff, the phase will switch to climb when the thrust revision altitude is achieved.
- Switch from **climb** to **cruise** (level):

$$|h_{cruise} - h| < K_{capture}\dot{h}$$



# Vertical guidance

## Vertical leg transition

- The vertical path is composed of several segments. Just as in the lateral domain it is desirable to anticipate the activation of the next vertical leg such that a smooth capture of that segment is performed without path overshoot.

$$K_{capture}\left|h_n^{path} - h_{n+1}^{path}\right| < \left|\dot{h}_n^{desired} - \dot{h}_{n+1}^{desired}\right|$$