ELEC 3224 — Guidance Navigation and Control of UAVs

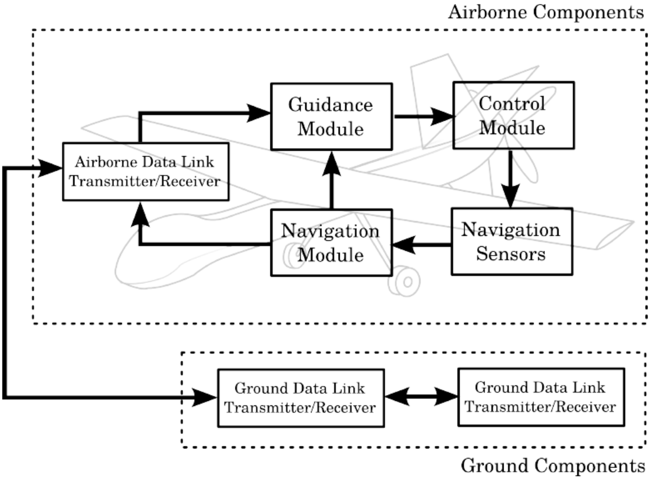
Professor Eric Rogers

School of Electronics and Computer Science
University of Southampton
etar@ecs.soton.ac.uk
Office: Building 1, Room 2037

Introduction

- A general architecture for this general area has been shown previously (given again in the next figure).
- In this section, the focus is on INS/GPS Navigation.
- Inner loop Control, and
- Guidance law.

Introduction



Introduction

- An inertial navigation system (INS) uses of the output of inertial sensors to estimate the vehicle's position, velocity, and attitude.
- A complete 6 DOF inertial sensor consists of 3-axis accelerometers and 3-axis gyros.
- The accelerometer measures the specific force acting on the platform and the gyros measure its rotation.
- Inertial sensors can be categorised according to the resulting navigational accuracy.

Introduction

- In some cases, when operating as a stand alone navigator such sensors produce large positioning errors. These errors in position, velocity and attitude estimates are mainly due to the corrupting effects of sensor noise.
- The position error **grows linearly with the initial velocity error estimate, quadratically with uncorrected bias and at a cubic rate with attitude error.**
- Why use them at all?
- (An) Answer: that they are **self-contained,** i.e., they require no external signal to provide a navigation solution. In terms of the data rate, the **navigation states can be computed at a rate limited only by the processing power of the flight computer.**

Introduction

- Why not use GPS?
- It can be shown that the GPS navigation has long-term stability, the bandwidth of the solution is much lower than that of an inertial navigator.
- **Integration of INS with GPS allows a navigation solution that has the high bandwidth of the inertial sensors and the drift-free long term stability of the GPS solution.**
- Attitude determination is an integral part of strapdown INS.
- This is because the specific force measured by the accelerometer has to be **transformed into the navigation frame in which the position and velocity are calculated. This, in turn, needs knowledge of the orientation of the platform, defined as the attitude.**

Attitude Determination

- Attitude can be equivalently described by a set of 3 angles — known as the **Euler angle sequence** or **the quaternion.**
- A Euler angle sequence consists of the yaw (ψ), pitch (θ) roll (ϕ) that **describe three successive rotations** about, respectively, the body axes z, y and x.
- This representation is physically motivated it is **singular at** $\theta = \pm 90^\circ$**.**
- A **quaternion is a set of four numbers that can be related to** the roll pitch and yaw angles using the following relationship:

Attitude Determination

- $$\phi = \tan^{-1}\left(\frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + 2q_3^2 - 1}\right)$$
- $$\theta = \sin^{-1}\left(-2q_1q_3 + 2q_0q_2\right)$$
- $$\psi = \tan^{-1}\left(\frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + 2q_3^2 - 1}\right)$$
- The quaternion $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$ **represents the translation from the local NED (North-East-Down) co-ordinates to the body-fixed frame.**

Attitude Determination

- Although it has no singularity, directly visualising the vehicles orientation can be challenging.
- Hence for remote pilot display and most control laws, the quaternion must be transformed to its corresponding roll, pitch and yaw angles.
- Attitude determination using gyros requires the angular velocity to be integrated to propagate the attitude forward in time.
- Gyros measure **inertial rotation and hence must be compensated to account for both the earths rotation and the transport rate due to the earths curvature.**
- For most UAVs, these are small compared to the noise level in the sensors and hence can be neglected for all practical purposes.

Attitude Determination

- To improve the navigation solution between subsequent GPS solutions, the filter makes frequent corrections to compensate for inertial sensor errors.
- Numerous models to represent these errors are available.
- In one such model the sensor output (e.g., a gyro) as a function of time is written as

$$\omega(t) = \overline{\omega}(t) + b_{gs} + b_{gd}(t) + \omega_g$$

- $\overline{\omega}(t)$ — true angular frequency, b_{gs} — turn-on bias, $b_{gd}(t)$ — a time-varying bias and ω_g is measurement noise — taken as white noise.

Attitude Determination

- $b_{gd}(t)$ is modeled as a first order Gauss-Markov process:

$$\dot{b}_{gd}(t) = -\frac{1}{\tau} b_{gd}(t) + \omega_b$$

- Using this model, the estimated sensor bias includes the true estimated bias but also accounts for all unmodeled errors that corrupt the sensor measurement.
- Next figure — architecture of the INS/GPS filter using an extended Kalman filter (EKF) – the state vector (Euler angle representation is

$$x = \left[\begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \right]^T$$

Attitude Determination

- X_1 — position

$$X_1^T = \begin{bmatrix} L \\ \Lambda \\ h \end{bmatrix}$$

- X_2 — ground speed

$$X_2^T = \begin{bmatrix} V_{North} \\ V_{East} \\ V_{Down} \end{bmatrix}$$

Attitude Determination

- X_3 – attitude

$$X_3^T = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}$$

- X_4^T – Accelerator Bias

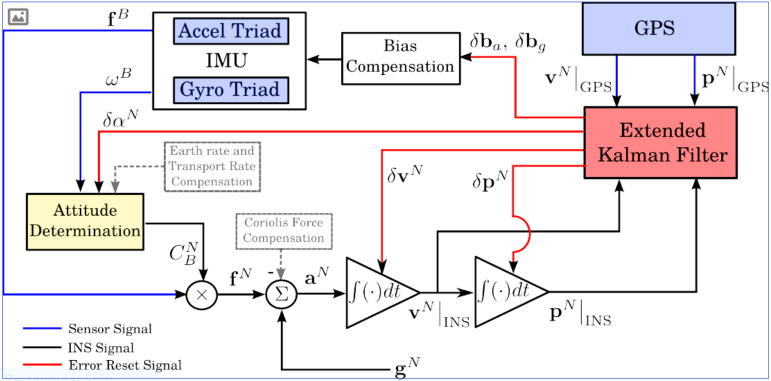
$$\begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix}$$

Attitude Determination

- X_5 – Gyroscope Bias

$$X_5^T = \begin{bmatrix} b_{gx} \\ b_{gy} \\ b_{gz} \end{bmatrix}$$

INS/GPS Integration



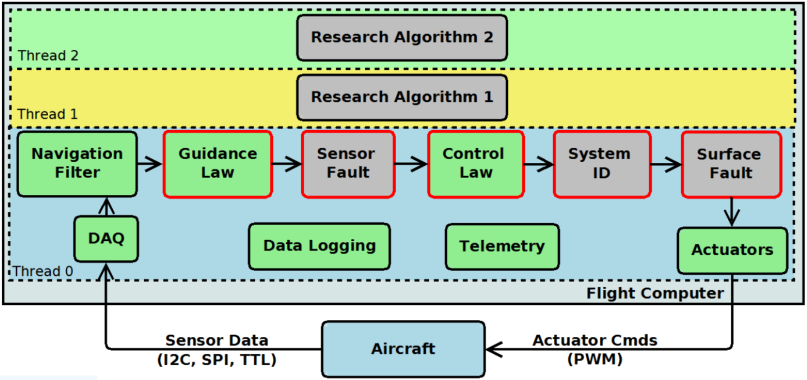
INS/GPS Integration — Software

- Needs a real-time operating system (RTOS).
- This computer handles data acquisition; performs guidance, navigation and control tasks; stores relevant data; and sends information to the ground control station via the telemetry modem.
- The flight software uses a multi-threaded architecture in which all of the flight critical tasks execute in the highest priority thread a specified frequency, while additional tasks, i.e., those not required to control the aircraft, such as a fault detection filter are executed in separate, lower priority threads.

INS/GPS Integration — Software

- As shown in the next figure, there are ten modules executed in the highest priority thread (thread 0).
- Navigation tasks are executed immediately after data acquisition (DAQ)
- It starts as soon as valid GPS measurements are available.
- The position and velocity are initialized at the reported position and velocity.
- Since this initialization occurs on the ground, attitude is initialized to a known attitude of the aircraft on the ground.

INS/GPS Integration — Software



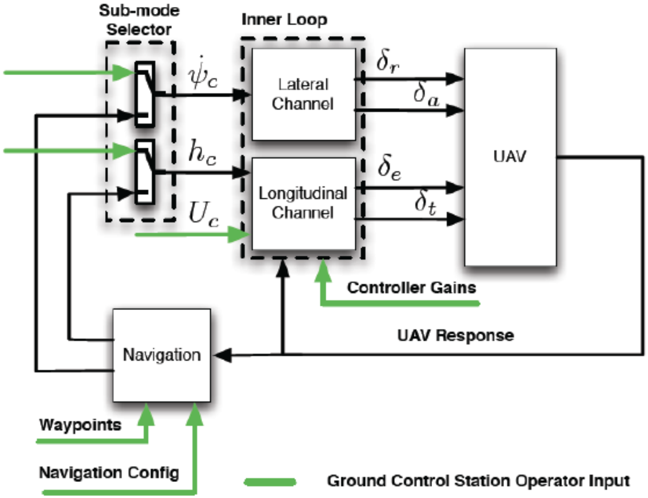
Inner Loop Control

- To fly an aircraft, a low level control system must stabilize the airframe using available sensor inputs and actuators.
- A higher level outer loop control will implement path following (see later) while the inner loop keeps the aircraft flying.
- There are many options to implement inner loop control for an AUV.
- Here an autopilot that uses PID controllers in the inner loop — this autopilot has been successfully flight tested.
- Block diagram of the inner loop and guidance — next figure.

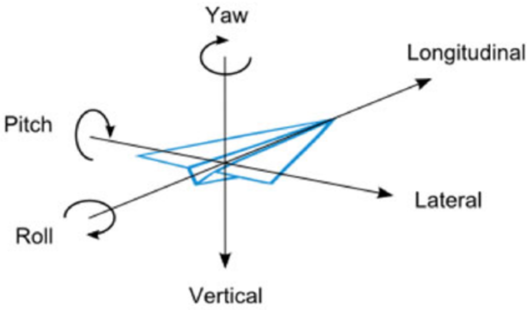
Inner Loop Control

- This autopilot is divided into two hardware sections: **a control processor and a sensor processor.**
- The sensor processor is tasked with taking the raw sensor measurements and fusing them into a high quality position and attitude estimate.
- The control processor is tasked with both the inner loop (stabilization) and the outer loop (guidance).

Inner Loop Block Diagram and Guidance



Recap



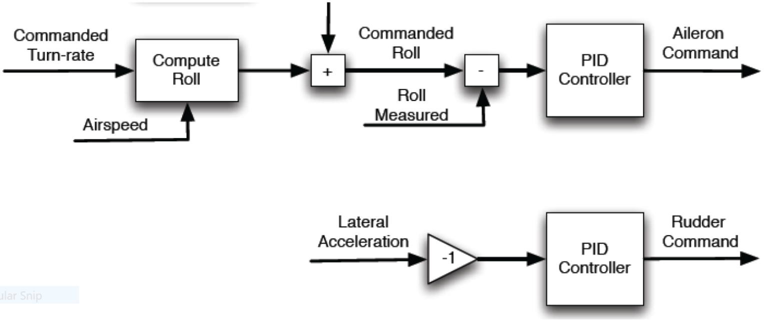
Inner Loop Block Diagram and Guidance

- Many possible designs.
- One is based on **decoupling** the longitudinal and lateral flight dynamics into **two separate** control systems.
- The lateral controller regulates the rudder and ailerons and the longitudinal controller regulates the throttle and elevator.
- **Note:** This form of decoupling control is appropriate for aircraft-like UAVS flying gentle maneuvers it is **not appropriate for cases where there is a high degree of lateral-longitudinal cross-coupling nor for aircraft performing aggressive maneuvers.**

Lateral Control

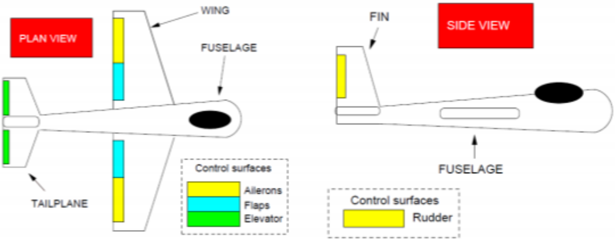
- The two main (lateral and longitudinal) controllers are divided into successive closure using PID controllers.
- **Lateral Control** — see the next figure.
- This controller uses the rudder and ailerons to keep the aircraft flying in a coordinated turn, and following a commanded turn rate (including a zero turn rate for straight flight).
- The lateral dynamics of an aircraft include the roll rate damping mode, a spiral mode, and the dutch roll mode (yaw-roll coupling).

Recap


Commanded
Turn-rate
Compute
Roll
Airspeed
+
Commanded
Roll
-
Roll
Measured
PID
Controller
Aileron
Command
Lateral
Acceleration
-1
PID
Controller
Rudder
Command

Recap

Structure of typical aircraft



Wing	Primary device for lift generation
Fuselage	Payload, guidance, control, communications
Elevator	primary device for altering aircraft pitch
Ailerons	Responsible for aircraft roll
Rudder	Responsible for aircraft yaw
Flaps	For increasing lift at low velocities
Turbofan/jet/propeller	Thrust generation

Lateral Control

- In this formulation of the inner loop control, the yaw rate $\dot{\psi}_c$ is commanded and converted to a command roll angle ϕ_c using

$$\phi_c = \tan^{-1}\left(\frac{\dot{\psi}_c U_m}{g}\right) \tag{1}$$

- U_m – measured speed, g – gravity.
- This equation assumes that the aircraft is flying a coordinated turn, i.e., the turn rate is constant and the body fixed lateral acceleration $a_y = 0$.
- The output of the PID controller is fed to the ailerons, which is used to drive the roll error, i.e., commanded - actual, to zero.

Lateral Control

- The commanded roll angle ϕ_c is used as the reference to the PID control, with the plant output being the actual roll angle, ϕ produced by the attitude estimation algorithm.
- The commanded bank angle is limited with a saturation block to keep the roll angle from becoming too large — possible limit $\pm 40^\circ$.
- In this case, the derivative of the roll error is taken directly from the body fixed gyros — with bias again accounted for by the attitude estimation algorithm.
- Hence

$$\dot{\phi} = P + (q \sin \phi + r \cos \phi) \tan \theta$$

Lateral Control

- where $\begin{bmatrix} p & q & r \end{bmatrix}$ are the body fixed roll, pitch and yaw rates.
- ϕ, θ are, respectively, the roll and pitch Euler angles.
- For a small pitch angle and yaw rate

$$\dot{\phi} \approx p$$

- When the roll command ϕ_c is of opposite polarity to the previous command, the integral state is reset to improve roll tracking performance.

Lateral Control

- The above desgin describes the turn rate command to the aileron command, which will drive the roll error to zero.
- However, (1) requires coordinated flight to be effective.
- To ensure coordinated flight, a second PID loop around the rudder is used.
- The commanded input to this PID controller is the **negative** of the lateral acceleration since the rudder the rudder is behind the center of mass of the aircraft, and has a negative sign in the transfer-function), and the output is the rudder actuator command.

Lateral Control

- This PID loop will actuate the rudder to drive the body fixed lateral acceleration a_y to zero.
- Again, the fact that the output of the aircraft affected by the rudder has a direct measurement of its derivative from the body fixed gyros can be used for the D term, i.e.,

$$\dot{\psi} = \frac{1}{\cos\theta}(q \sin\phi + r \cos\phi)$$

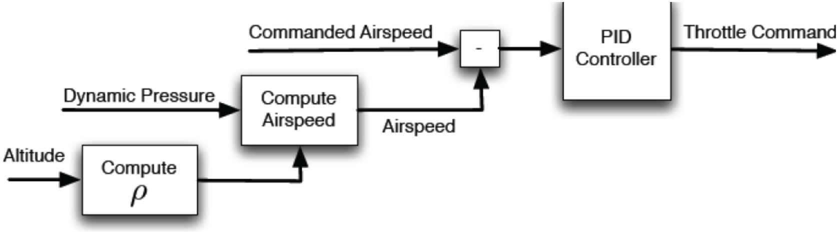
- or, for small angles of pitch and roll

$$\dot{\psi} \approx r$$

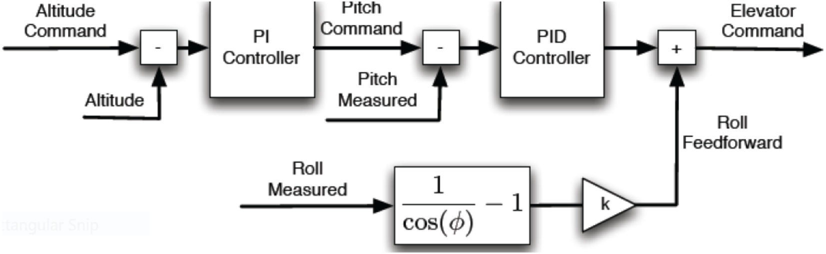
Longitudinal Control

- The low level longitudinal loops — see the next 2 figures — operate in a similar manner to the lateral loops, with the two flight controls — **the throttle and elevator.**
- Two modes of operation: **climb/descent** and **level flight.**

Longitudinal Control



Longitudinal Control



Longitudinal Control

- In level light mode, airspeed is **controlled directly by the throttle and altitude by aircraft pitch via the elevator.**
- Since airspeed is being kept constant, the altitude responds rapidly to even small changes in aircraft pitch.
- As a design choice, only the level light mode is implemented, given that the UAV will spend most of its mission in level flight at constant altitude following waypoints.

Longitudinal Control

- The airspeed hold control loop takes in a commanded airspeed, U_c and compares it to the measured airspeed.
- The airspeed control uses airspeed, and not ground speed (as available from GPS).
- Compensation for wind is handled within the higher level guidance controller, rather than in the low level stabilizing controllers.
- The vehicle does not have onboard measurement of airspeed. Instead, onboard measurement of dynamic pressure q and altitude.
- Then

$$U_m = \sqrt{\frac{2q}{\rho}}$$

Longitudinal Control

- Also ρ is the atmospheric density, which is a function of altitude, where

$$\rho = \rho_0 (1 - \frac{h_m}{44331})^{4.255876}$$

where $\rho_0 = 1.025 Kg/m^3$ is the sea-level density and h_m is the measured altitude in meters.

- This approximation is valid for most heights reached by UAVs and for control.
- The PID output is the throttle actuator signal

Longitudinal Control

- In the case of the airspeed, there is no direct derivative to be used in the throttle loop (though the body fixed longitudinal acceleration could be used).
- Care must be used in selecting the derivative terms and in tuning the airspeed hold loop.
- Too aggressive gains on the airspeed hold loop causes the throttle to surge and cut in flight — not good.
- This is due both to the noise on measured airspeed and also due to the lag in response to the throttle input. Here tuning the gains for an acceptable error and relying on the integral term to close the error works well.

Longitudinal Control

- The altitude control loop consists of two chained PID controllers.
- The first is PI, where the error used is the difference between the commanded altitude h_c and the measured altitude h_m and the output is a pitch command.
- The pitch command is saturated at $\pm 15^\circ$.
- The resulting limited pitch command is the reference signal for the second PID loop.

Longitudinal Control

- The limited pitch command is comapred with the UAV pitch, θ, from the attitude estimation to form the error.
- The output of the second PID loop drives the elevator to match the commanded and actual UAV pitch angle.
- In this case the derivative term is obtained direct from the body fixed gyros, i.e.,

$$\dot{\theta} = q \cos \phi - r \sin \phi$$

- Or for small bank angles

$$\dot{\theta} \approx q$$

Longitudinal Control

- The direct derivative term allows the pitch to elevator PID control to be aggressive, yet still retain good damping characteristics.
- This PID controller will drive the aircraft pitch to the commanded pitch until the desired altitude is reached.
- While the aircraft is flying straight and level, this control works very well.
- However, while the aircraft is in a turn, the aircraft will descend. This is because the lift from the wings must match the aircraft weight, but in a turn part of the lift is directed inwards to cause the turn itself.

Longitudinal Control

- To avoid losing altitude during a turn, the lift is related to the weight and bank angle as

$$L = \frac{W}{\cos\phi}$$

where L is the lift, W is the weight and ϕ is the roll angle.

- Simplified case — lift only from the wings etc, the change in the lift is

$$\Delta L = W\left(\frac{1}{\cos\phi} - 1\right)$$

which says how much the lift must be increased to maintain altitude in a coordinated circular turn.

Longitudinal Control

- For piloted flight — the pilots are **trained to increase the elevator (back stick) when entering a turn to maintain altitude.**
- For UAVs, to maintain altitude during turns a feedforward gain proportional to the increase in lift is included:

$$\delta_{eff} = K_{ff}\left(\frac{1}{\cos\phi} - 1\right)$$

where δ_{eff} is the additional elevator command due to the feedforward term.

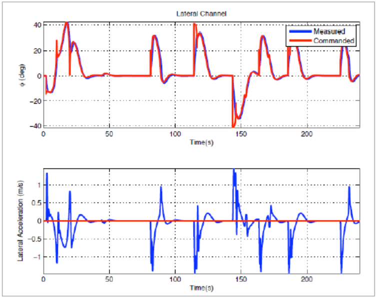
Longitudinal Control

- In implementation, the roll angle, ϕ, is low-pass filtered to reduce unwanted pitch oscillations resulting from attitude estimation noise, and is also limited to $\pm 60^\circ$.
- At greater angles, the UAV will simply enter into an accelerated stall trying to hold altitude and the low level control system will no longer be able to stabilize the aircraft.

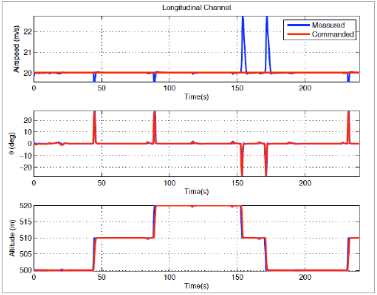
Trim Conditions

- By definition, to **an aircraft is to adjust the aerodynamic forces on the control surfaces so that the aircraft maintains the set attitude without any control input.**
- While all axes of rotation are affected by aerodynamic forces, not all aircraft types are capable of being trimmed in all three axes.
- Virtually all aircraft designs incorporate some form of pitch axis trim and most have provisions of some description for trimming in the yaw axis.
- Roll axis trim exists on many aircraft but it is the least frequently encountered installation of the three.
- There are several different types of trim systems in use and more than one type may be found on a given aircraft.

A Simulation Study



(a)



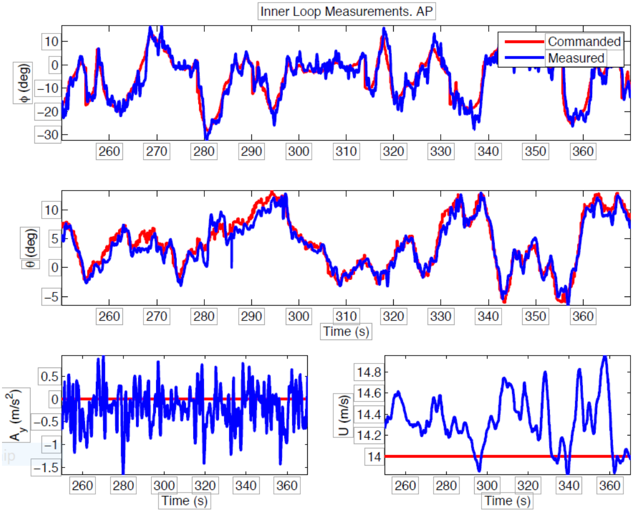
(b)

- a) the lateral channel, b) the longitudinal channel.

Simulation and Flight Test Results

- Previous figure — data from a simulation study based on a 'realistic model' for the inner loop PID controllers.
- Every change in commanded roll angle produces a large disturbance change in the lateral acceleration, which must be washed out (a washout filter not covered here). In panel (b) is the longitudinal control, with both pitch and airspeed control.
- Again, the spikes out of speed are due to large climb or descent inputs to the controller.

Flight Test Results (Not done in Southampton)



Flight Test Results (Not done in Southampton)

- Previous figure — flight test data from the inner loop control, which shows excellent performance on pitch and roll commands, a reasonable attenuation of lateral acceleration (with a limit to less than 1g for most of the flight time), and a fairly good airspeed hold.
- This is in the presence of wind, gusts, and other disturbances.
- The bias on the lateral acceleration is most likely from the accelerometer being mounted with a slight tilt on the airframe.

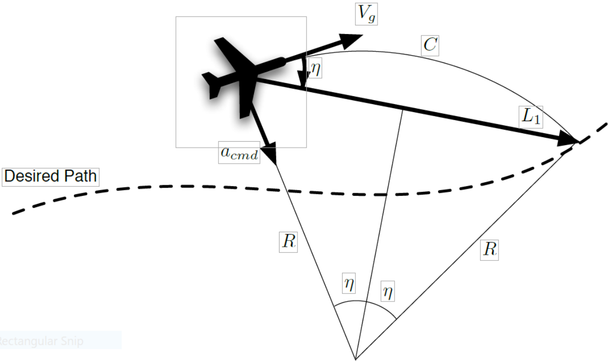
Guidance

- The SLUGS autopilot deliver a mission to the UAV from the ground station, a simple set of GPS-based waypoints, along with an altitude and speed for each leg of the mission, is transmitted to the UAV via telemetry link.
- The interest is in flying the vehicle on the trajector and therefore the **legs are blended into each other using circular arcs rather than forcing the UAV to overly each waypoint.**
- The SLUGS guidance algorithm is based **on an extension of a simple IOS guidance law originally developed for ground robotics. Much of the art of waypoint guidance consists of determining which leg of the trajectory the UAV is on, and when to switch to the next leg of the trajectory.**

Guidance

- The outer loop guidance law steers the velocity vector toward the line of sight — this is **one form of pursuit guidance.**
- Setting the commanded acceleration proportional to $\sin \eta$ — see the next figure — is one of number of possible guidance laws.
- The basic guidance algorithm is to determine an aim point and steer the velocity vector towards it.

Guidance



Rectangular Snip

Guidance

- In the previous figure — V_g is the UAV groundspeed and C is a circular arc of radius R, which originates at the UAV and intercepts the desired path and L_1 is a constant lookahead distance from the UAV to the path in the desired direction of travel.
- By basic trigonometry

$$\frac{|L_1|}{2} = R \sin \eta \tag{2}$$

- The **centripetal acceleration,** a_c, required be a point mass to follow a circular arc C is

$$a_c = \frac{|V_g|^2}{R} \tag{3}$$

Guidance

- Hence the UAV must command a lateral acceleration a_c. Solving (2) for R and substituting in (3) gives the commanded acceleration

$$a_{cmd} = 2\frac{|V_g|^2}{|L_1|}\sin\eta \tag{4}$$

- To implement this control law it is required to select the lookahead distance $|L_1|$ and determine $\sin\eta$ — **the sine of the angle from the velocity vector to L_1.**
- η is also referred to the **line of sight angle.**
- Choosing $|L_1|$ is analogous to selecting a feedback gain in control — **larger L_1 corresponds to smaller gains.**

Guidance

- Also

$$\sin \eta = \frac{V_g \times L_1}{|V_g||L_1|} \tag{5}$$

- $\times$ — vector product.
- For the UAV to actually track the desired trajectory, the lateral acceleration command must be converted to an appropriate bank angle command using the steady-state turn equation

$$\phi = \tan^{-1} \frac{a_{cmd}}{g}$$

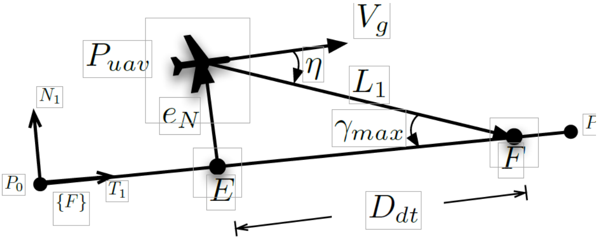
Guidance — Straight Line Tracking

- In the next figure the UAV is following a straight line segment from waypoint P_0 to P_1.
- In this case

$$T = \left[\begin{array}{ccc} x_T & y_T & z_T \end{array} \right] = \frac{(P_1 - P_0)}{|P_1 - P_0|}$$

$$N = \frac{1}{\left| \left[\begin{array}{ccc} -y_T & x_T & 0 \end{array} \right] \right|} \left[\begin{array}{ccc} -y_T & x_T & 0 \end{array} \right]^T$$

Guidance — Straight Line Tracking



Guidance — Straight Line Tracking

- $$B = T \times N$$
- Also e_N is the error in the N direction and hence
$$|e_N| = N^T(P_{uav} - P_0)$$
- E is the closest point on the path to the UAV.
- The intersection of L_1 with the desired path is marked by the point F.

Guidance — Straight Line Tracking

- The **down track distance** or EF separation, D_{df}, is given by

$$D_{dt} = \sqrt{|L_1|^2 - |e_N|^2}$$

- also

$$\begin{aligned} E &= P_{uav} - |e_N|N \\ F &= E + D_{dt}T \end{aligned}$$

- Hence

$$L_1 = F - P_{uav} = D_{dt}T - |e_N|N$$

Guidance — Straight Line Tracking

- This last formula can be used in (5) to determine η and hence the commanded acceleration.
- For sufficiently small tracking errors, L_1 is similar to a PD controller acting on the lateral error.
- This autopilot extends the L_1 pursuit guidance to account for some shortcomings of the control law — resulting in L_2^+ control.
- It was noticed during flight test experiments that the L_1 guidance exhibited large overshoots when turning downwind (hence an increasing V_g).
- Analysis showed that to solve this problem, the new L_2 vector should **be a function of groundspeed.**

Guidance — Straight Line Tracking

- Hence

$$|L_2| = T^*|V_g|$$

where T^* is a constant and the commanded acceleration becomes

- $$a_{cmd} = 2\frac{|V_g|}{T^*}\sin\eta$$

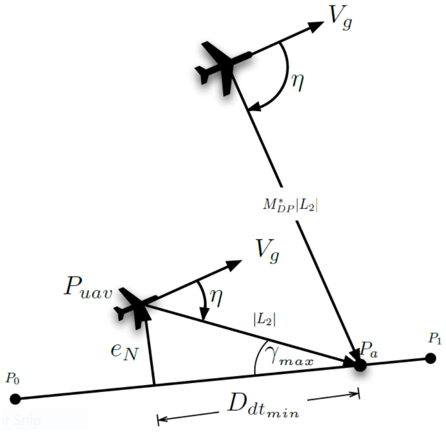
Guidance — Straight Line Tracking

- When the lateral error$|e_N| > |L_2|$, the aim point cannot be found.
- To ensure that an aim point is always found requires a maximum intercept angle γ_{max} to be defined.
- The downtrack distance to the aim point defined by γ_{max} is

$$D_{dt} = \frac{|e_N|}{\tan \gamma_{max}}$$

- see the next figure (bottom).

Guidance — Straight Line Tracking



- L_2^+ geometry.

Guidance — Straight Line Tracking

- When $|e_N| > |L_2|$, e.g., during an initial intercept, D_{dt} may be too large, e.g., the aim point may go beyond the next waypoint.
- To counter this, a bound is placed on the down path distance of the aim point. This bound is set as constant, $\mathcal{M}^*_{DP}$, times $|L_2|$

$$D_{dt_{min}} = \min(D_{at}, |L_2|\mathcal{M}^*_{DP})$$

- see the previous figure (top).
- The down track distance of the aim point is

$$D^*_{dt_{min}} = \begin{cases} D_{dt_{min}}, & |e_N| > |L_2| \\ \max(D_{dt_{min}}, \sqrt{|L_2|^2 - |e_N|^2}, & |e_N| < |L_2| \end{cases}$$

Guidance — Straight Line Tracking

- If the aim point is beyond P_1, the UAV will continue along this line without changing direction.
- Let D_{wp_1} be the along the track distance from the UAV to P_1 is

$$D_{wp_1} = T^T(P_1 - P_{uav})$$

  and is negative if the UAV is beyond P_1.
- The distance from P_1 back along T to the aim point is

$$D_a = D_{wp_1} - D^*_{dt_{min}}$$

Guidance — Straight Line Tracking

- The max operation ensures that the aim point does not extend beyond P_1, i.e., if $D_a < 0$ due to UAV position or failure of the waypoint switching logic.
- The acceleration command is then given by (4) with $\sin\eta$ given by the cross product of V_g and P_a. (5))
- If the UAV is initially pointed in the opposite direction from YT, $\eta > 90^\circ$.
- In this case, the maximum lateral acceleration is used to return the UAV to the correct flight, where

$$a_{max} = g \tan \phi_{max}$$

Guidance — Straight Line Tracking

- ϕ_{max} is the maximum back angle permitted — $30 - 60^\circ$ for a small UAV.
- To use the $L_2^=$ controller in this case a maximum allowable η is required

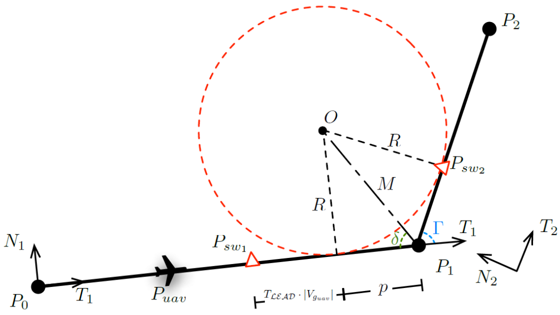
$$\eta_{max} = \min\left(\frac{\pi}{2}, T_{Lead}\frac{a_{max}}{U_{comm}}\right)$$

  where U_{comm} is the commanded airspeed from the inner loop controller and T_{Lead} is the lead time required for the UAV to ramp out of a steep bank — determined experimentally for the airframe.

Guidance — Waypoint Switching

- The guidance strategy is to move from one segment to the next of the flight mission by connecting the two segments with a circular arc. An L^{+}_{The2} implementation uses early waypoint switching to give priority to path-following over precision in the waypoint.
- In the next figure — U_c — commanded airspeed, wind — the wind speed, $a_{\max}$ — maximum acceleration of the UAV

Guidance — Waypoint Switching



Guidance — Waypoint Switching

- Let C be a circle of radius R and given by

$$R = \frac{(U_c + |wind|)^2}{a_{max}}$$

- where
- U_c is the commanded air speed.
- $|wind|$ is the wind speed.
- a_{max} is the maximum acceleration for the UAV.

Guidance — Waypoint Switching

- In the presence of wind the actual radius of curvature of the vehicle over the ground changes as the track angle changes.
- To avoid this problem in defining switching points, the maximum possible ground speed should be used in the numerator, which creates a circle with radius larger than any curve over the ground.
- The first tangent point, P_{sw1}, in the last figure determines the location where the UAV will switch to start tracking the next waypoint segment.

Guidance — Waypoint Switching

- $$\Gamma = \arccos T_1^T T_2$$
  $$\delta = \frac{\Gamma - \pi}{2}$$
  Also
  $$p = M\cos\delta, \; : M = \frac{R}{\sin\delta}$$
  Hence
  $$p = \frac{R}{\tan\delta}$$

Guidance — Waypoint Switching

- Initiating the turn just at the switch point given by the last equation is not adequate due to the lag time of the roll dynamics of the UAV. To counter this a lead time, say τ_{lead} can be introduced to initiate the turn. Multiplying τ_{lead} by the groundspeed, gives the extra distance from the waypoint to the switch point.
- The new switch point distance is

$$p = \tau_{\text{lead}}|V_g| + \frac{R}{\tan\delta}$$

and

$$P_{\text{sw1}} = P_1 - pT_1$$

Guidance — Waypoint Switching

- During the transition from missions, the L_2 vector intercepts the circular arc, not the straight line segments. Also, depending on the exact geometry of the waypoints, and the aircraft speed, it may be that just after switching legs, the aircraft is already beyond the next segment switching point. In this case, the logic immediately switches again to the next leg.
- **Point Acquisition and RTB**
- The L_2^+ controller is robust, e.g., without any change in logic the same controller can be used to drive the UAV to a point in addition to an arbitrary curve.

Guidance — Waypoint Switching

- For acquisition point P_a,

$$\sin\eta = \frac{V_g \times (P_a - P_{uav})}{|V_g||(P_a - P_{uav})|}$$

- Again with the limits η_{max} and the downrange distance imposed to limit bank angle and lateral acceleration apply.
- This controller will essentially 'point at the point' until it overflies the acquisition point, P_a, at which point the switch logic will cause it to circle the point.
- Nowhere in the formulation does the point P_a have to be fixed — the same L_2^+ controller can track a moving target using its same logic given a position estimate of the target.
- Simulations have confirmed that for target speeds moving at or below the ground speed of the UAV, the target is always acquired.

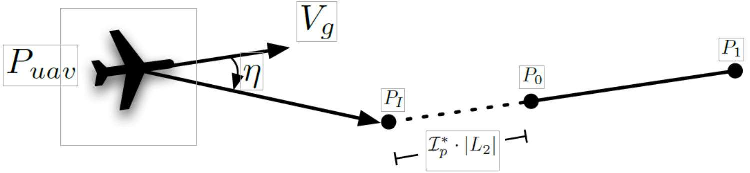
Guidance — Waypoint Switching

- The initial mission point, P_1, before the first waypoint, is determined using a concept borrowed from instrument flying in which all aircraft must first fly to a well defined point before proceeding to land.
- The initial point is determined by projecting a fixed point in front of the first leg of the mission a constant distance in front of the initial waypoint — see next figure.

$$P_I = P_a = P_0 - T_1(I_p^* |L_2|)$$

- I^* — a constant tuned to the specific UAV.

Guidance — Waypoint Switching



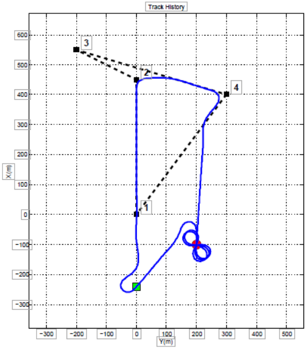
Guidance

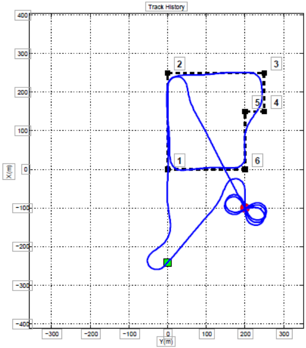
- Lastly, the L_2^+ controller implements a return to base (RTB) functionality by simply recording the original base position, P_b and using this as the acquisition point if either the mission waypoints are completed or communication to the UAV fails.

Simulation and Flight Test Results

- Both simulation, **Hardware-in-the-Loop** tests and actual flight tests have been undertaken (not in Southampton).
- UAV simulated using a full 6 DOF description.
- Next figure — L_2^+ controller running — showing initial point, transitions and a RTB at the end of the flight.

Simulation and Flight Test Results





Simulation and Flight Test Results

- Both cases have an initial point (green square) and RTB (red square).
- In the left-hand plot, the logic skips the waypoint 2 and in the right-hand plot the waypoint 5.
- Next figure left plot — results in the presence of strong wind and real flight disturbances.
- Next figure right plot — L_2 vector in real time as the UAV transitions through the waypoints.

Simulation and Flight Test Results