A General Solution to the Aircraft Trim Problem

Agostino De Marco[1] Eugene L. Duke[2] Jon S. Berndt[3]

[1]University of Naples "Federico II"
Department of Aerospace Engineering (DIAS)
Via Claudio, 21 – 80125 Naples, Italy

[2]Rain Mountain Systems Incorporated
Glasgow, Virginia, 24555-2509, USA

[3]JSBSim Project Team Lead
League City, TX, USA

AIAA Modelling & Simulation Technology Conference
Hilton Head, South Carolina, 2007

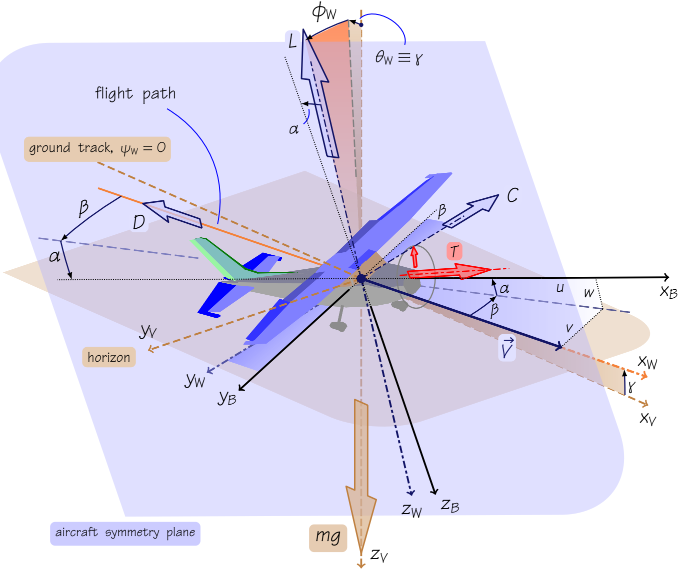
Outline

Outline

The aircraft trim problem: our approach

- We define what we mean by trim.
- Examine a variety of trim conditions.
- Derive the equations defining those trim conditions.
- Present a general approach to the solution of trim problems: constrained minimization of a multivariate, scalar cost function.
- Provide an example of how trim algorithms are applied in JSBSim, an open source flight dynamics model.

Definitions: frames, aerodynamic angles, forces



Definitions: state space concepts

- Generic aircraft state equations

implicit form: $\boldsymbol{g}\big(\dot{\boldsymbol{x}},\, \boldsymbol{x},\, \boldsymbol{u}\big) = 0$ explicit form: $\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x},\, \boldsymbol{u}\big)$

$\boldsymbol{x} =$ aircraft state vector, $\boldsymbol{u} =$ vector of inputs.

- *Dynamic* & *kinematic* parts

$$\boldsymbol{x} = \big[\, \boldsymbol{x}_d^{\mathrm{T}} \,,\, \boldsymbol{x}_k^{\mathrm{T}} \,\big]^{\mathrm{T}}$$

- Body-fixed frame components, Earth-frame c.g. coordinates and A/C Euler angles

$$\boldsymbol{x}_d = \big[\, u \,,\, v \,,\, w \,,\, p \,,\, q \,,\, r \,\big]^{\mathrm{T}} ,\quad \boldsymbol{x}_k = \big[\, x_C \,,\, y_C \,,\, x_C \,,\, \phi \,,\, \theta \,,\, \psi \,\big]^{\mathrm{T}}$$

Definitions: inputs

- Vector $\boldsymbol{u}$ of control inputs depends by the type of aircraft.
- For a conventional configuration aircraft the minimum arrangement of the inputs is usually given by

$$\boldsymbol{u} = \left[\, \delta_T \,,\, \delta_e \,,\, \delta_a \,,\, \delta_r \,\right]^{\mathrm{T}}$$

- Inputs u_j have standard signs. Their ranges depend on the particular aircraft. Might be mapped to $[-1,+1]$ or to $[0,+1]$.
- We will always refer to the inputs as some required combination of aerosurface deflections and thrust output.

Steady-state conditions. Trim points. Equilibrium.

- A classical concept in the theory of nonlinear systems is the equilibrium point or trim point.
- For an autonomous, time-invariant system the equilibrium point is denoted as $\boldsymbol{x}_{\text{eq}}$ and is defined as the particular $\boldsymbol{x}$ vector which satisfies

$$\boldsymbol{g}\big(\boxed{0},\, \boldsymbol{x}_{\text{eq}},\, \boldsymbol{u}_{\text{eq}}\big) = 0 \quad \text{with } \boxed{\dot{x} \equiv 0} \text{ and } \boldsymbol{u}_{\text{eq}} = \text{constant}$$

- Equilibrium condition above also defines a set of control settings $\boldsymbol{u}_{\text{eq}}$ that make the steady state possible.
- Generalized idea of rest: the condition when all the derivatives are identically zero. In our case this applies to the dynamic part $\boldsymbol{x}_d$, i.e. to those independent variables ruled by Newton's laws: $\dot{\boldsymbol{x}}_d \equiv 0$.

General conditions for steady-state flight

The general steady-state flight condition resulting from the above discussion is given as follows:

$$\text{zero accelerations} \Rightarrow \dot{u}, \dot{v}, \dot{w} \left(\text{or } \dot{V}, \dot{\alpha}, \dot{\beta}\right) \equiv 0$$
$$\dot{p}, \dot{q}, \dot{r} \left(\text{or } \dot{p}_\text{W}, \dot{q}_\text{W}, \dot{r}_\text{W}\right) \equiv 0\,,$$
$$\text{linear velocities} \Rightarrow u, v, w \,(\text{ or } V, \alpha, \beta)$$
$$= \text{constant values}\,,$$
$$\text{angular velocities} \Rightarrow p, q, r \,(\text{ or } p_\text{W}, q_\text{W}, r_\text{W})$$
$$= \text{prescribed/constrained constant values}$$
$$\text{aircraft controls} \Rightarrow \delta_T, \delta_e, \delta_a, \delta_r$$
$$= \text{appropriate constant values}$$

Equations of aircraft motion: flat-Earth hypothesis

- Round-Earth equations can be relaxed to flat-Earth equations.
- Usually the flat-Earth equations are considered satisfactory for all control system design purposes. Consequently those equations are satisfactory also for the derivation of trim conditions.
- Allowed steady-state conditions: *wings-level horizontal flight*, in any direction, and *constant altitude turning flight*.
- If the change in atmospheric density with altitude is neglected, also *wings-level climb* and *climbing turn* are permitted as steady state flight conditions.

Outline

System of nonlinear algebraic equations. Multiple solutions.

- An actual pilot may not find it very difficult to put an aircraft into a steady-state flight condition.
- Trimming an aircraft mathematical model requires the solution of simultaneous nonlinear equations: $\boldsymbol{g}(0,\, \boldsymbol{x}_{\mathrm{eq}},\, \boldsymbol{u}_{\mathrm{eq}}) = 0$.
- A steady-state solution can only be found by using a numerical method. Multiple feasible trim solutions for a given trim condition are possible.

A general approach. Aerodynamic database required.

- The treatment of aircraft trim proposed here starts from the standard 6DOF EOM of an airplane in atmospheric flight but does *not* make any limiting assumptions on the geometrical properties of the aircraft nor on the curves of aerodynamic coefficient
- All that one really expects is an aerodynamic model that provides non-dimensional aerodynamic coefficients, no matter where those parameters come from or how they are derived.
- In practice, a convenient aerodynamic model should be available in the form of tabulated data for the widest possible ranges of aerodynamic angles and velocities and for all possible aircraft configurations.

Outline

Aircraft equations of motion – Translation $(u,\, v,\, w)$

The classical system of scalar force equations in the body-axis reference

$$\begin{cases} \dot{u} = \dfrac{1}{m}\left(X_\mathrm{A} + X_\mathrm{T}\right) - wq + vr - g\sin\theta \\ \dot{v} = \dfrac{1}{m}\left(Y_\mathrm{A} + Y_\mathrm{T}\right) - ur + wp + g\cos\theta\sin\phi \\ \dot{w} = \dfrac{1}{m}\left(Z_\mathrm{A} + Z_\mathrm{T}\right) - vp + uq + g\cos\theta\cos\phi \end{cases}$$

$$V = \sqrt{u^2 + v^2 + w^2}$$

$$u = V\cos\beta\cos\alpha\,, \quad v = V\sin\beta\,, \quad w = V\cos\beta\sin\alpha$$

Aircraft equations of motion – Translation $(V,\,\alpha,\,\beta)$

$$
\left\{
\begin{aligned}
\dot{V} &= \frac{1}{m}\Big[-D\cos\beta + C\sin\beta + X_\mathrm{T}\cos\alpha\cos\beta + Y_\mathrm{T}\sin\beta + Z_\mathrm{T}\sin\alpha\cos\beta \\
&\qquad - mg\big(\sin\theta\cos\alpha\cos\beta - \cos\theta\sin\phi\sin\beta \\
&\qquad\qquad - \cos\theta\cos\phi\sin\alpha\cos\beta\big)\Big] \\
\dot{\alpha} &= q - \tan\beta\,(p\cos\alpha + r\sin\alpha) \\
&\quad + \frac{1}{Vm\cos\beta}\Big[-L + Z_\mathrm{T}\cos\alpha - X_\mathrm{T}\sin\alpha \\
&\qquad + mg\big(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\big)\Big] \\
\dot{\beta} &= +p\sin\alpha - r\cos\alpha \\
&\quad + \frac{1}{Vm}\Big[D\sin\beta + C\cos\beta - X_\mathrm{T}\cos\alpha\sin\beta + Y_\mathrm{T}\cos\beta - Z_\mathrm{T}\sin\alpha\sin\beta \\
&\qquad + m\,g\,\big(\sin\theta\cos\alpha\sin\beta + \cos\theta\sin\phi\cos\beta \\
&\qquad\qquad - \cos\theta\cos\phi\sin\alpha\sin\beta\big)\Big]
\end{aligned}
\right.
$$

Aircraft equations of motion – Rotation $(p,\,q,\,r)$

$$
\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = \frac{1}{\det\,[I]_{\mathrm{B}}} \begin{bmatrix} I_1 & I_2 & I_3 \\ I_2 & I_4 & I_5 \\ I_3 & I_5 & I_6 \end{bmatrix} \cdot \left(\begin{Bmatrix} \mathcal{L}_{\mathrm{A}} + \mathcal{L}_{\mathrm{T}} \\ \mathcal{M}_{\mathrm{A}} + \mathcal{M}_{\mathrm{T}} \\ \mathcal{N}_{\mathrm{A}} + \mathcal{N}_{\mathrm{T}} \end{Bmatrix} \right.
$$

$$
- \sum_k I_k^r\, \dot{\omega}_k^r \begin{Bmatrix} i_x^r \\ i_y^r \\ i_y^r \end{Bmatrix}_k
$$

$$
- \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \cdot \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \cdot \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}
$$

$$
\left. - \sum_k I_k^r\, \omega_k^r \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \cdot \begin{Bmatrix} i_x^r \\ i_y^r \\ i_z^r \end{Bmatrix}_k \right)
$$

Auxiliary kinematic equations

Navigation equations:

$$\begin{Bmatrix} \dot{x}_\text{E} \\ \dot{y}_\text{E} \\ \dot{z}_\text{E} \end{Bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \big(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\big) & \big(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\big) \\ \cos\theta\sin\psi & \big(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\big) & \big(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\big) \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

Gimbal equations:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & \dfrac{\sin\phi\sin\theta}{\cos\theta} & \dfrac{\cos\phi\sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \cdot \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}$$

Constraints

$$\begin{Bmatrix} p_W \\ q_W \\ r_W \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\gamma \\ 0 & \cos\phi_W & \cos\gamma\sin\phi_W \\ 0 & -\sin\phi_W & \cos\gamma\cos\phi_W \end{bmatrix} \begin{Bmatrix} \dot\phi_W \\ \dot\gamma \\ \dot\psi_W \end{Bmatrix}$$

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \left[C_{B\leftarrow W}\right] \begin{Bmatrix} p_W \\ q_W \\ r_W \end{Bmatrix} = \begin{Bmatrix} p_W\cos\alpha\cos\beta - q_W\cos\alpha\sin\beta - r_W\sin\alpha \\ p_W\sin\beta + q_W\cos\beta \\ p_W\sin\alpha\cos\beta - q_W\sin\alpha\sin\beta + r_W\cos\alpha \end{Bmatrix}$$

Example of constraint equations for a steady-state turn:

$$\dot\gamma = \dot\phi_W = 0, \qquad \dot\psi_W = \frac{g}{V}\tan\phi_W = \frac{g}{V}\tan\left[\frac{\left(n^2 - \cos^2\gamma\right)^{\frac{1}{2}}}{\cos\gamma}\right]$$

$$p = -\frac{g}{V}\left[\sin\gamma\tan\phi_W\cos\alpha\cos\beta + \frac{\cos\gamma\sin^2\phi_W\cos\alpha\sin\beta}{\cos\phi_W} + \cos\gamma\sin\phi_W\sin\alpha\right]$$

$$q = -\frac{g}{V}\left[\sin\gamma\tan\phi_W\sin\beta - \frac{\cos\gamma\sin^2\phi_W\cos\beta}{\cos\phi_W}\right]$$

$$r = -\frac{g}{V}\left[\sin\gamma\tan\phi_W\sin\alpha\cos\beta + \frac{\cos\gamma\sin^2\phi_W\sin\alpha\sin\beta}{\cos\phi_W} - \cos\gamma\sin\phi_W\cos\alpha\right]$$

Aerodynamics models

Table 1: Dependency of force and moment components upon aircraft state and control variables

Component		State Variables												
		h	V	α	β	p	q	r	$\dot{\alpha}$	$\dot{\beta}$	δ_e	δ_a	δ_r	δ_T
1	$X_A + X_T$	○[d]/●[e]	●	●	○	∼	∼	∼	∼	∼	○	∼	∼	●
2	$Y_A + Y_T$	○[d]/●[e]	●	○[a,b]	●	○	∼	●	∼	●/○	∼	○	●	○[c]
3	$Z_A + Z_T$	●[d,e]	●	●	○	○	●	∼	●	∼	●	∼	∼	○
4	$\mathcal{L}_A + \mathcal{L}_T$	○[d]/●[e]	●	○[a]	●	●	∼	●	∼	∼	∼	●	●	○
5	$\mathcal{M}_A + \mathcal{M}_T$	●[d,e]	●	●	○	∼	●	∼	●	∼	●	∼	∼	●/○
6	$\mathcal{N}_A + \mathcal{N}_T$	○[d]/●[e]	●	○	●	○	∼	●	∼	●/○	∼	○	●	○[c]

symbols: ● = dependent on
○ = could depend on, may vary with configuration
∼ = almost always not dependent on

[a] When β is not zero.

[b] When thrusters work in nonaxial flow.

[c] When a nonsymmetric thrust is applied.

[d] When ground effect is modelled.

[e] When engine state is altitude dependent.

Outline

Trim cost function

- Aircraft model equations: $\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x},\, \boldsymbol{u}\big)$

$$\text{Steady-state} \Rightarrow \dot{\boldsymbol{x}}_d \equiv 0 = \boldsymbol{f}_d\big(\boldsymbol{x}_d,\, \boldsymbol{x}_k,\, \boldsymbol{u}\big)$$

- Possible definitions of a cost function:

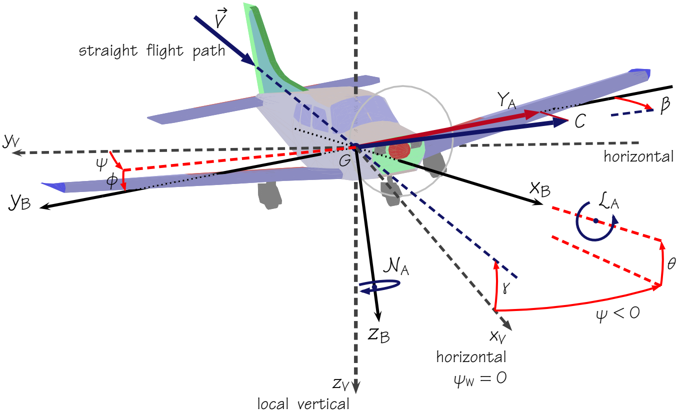
$$J = \dot{u}^2 + \dot{v}^2 + \dot{w}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2$$
$$J = \dot{V}^2 + \dot{\alpha}^2 + \dot{\beta}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2$$

- More generally $J = \{\dot{x}_d\}^{\mathrm{T}}[W]\{\dot{x}_d\}$, with $[W]$ a matrix of weights.
- **Aircraft steady-state flight yields a $J = 0$.**
- For a given, desired trim condition: Who are the variables upon which J depends?

Stating a trim problem

- Formally: $J = J\big(\boldsymbol{x}_d, \boldsymbol{x}_k, \boldsymbol{u}\big)$
  i.e. $J = J\big(u, v, w, p, q, r, h, \phi, \theta, \psi, \delta_T, \delta_e, \delta_a, \delta_r\big)$
- A trim problem is stated by:
  (*i*) declaring the desired center of mass trajectory, e.g. assigning V, γ, ψ_{GT}, h, $\dot{\psi}_{\mathrm{GT}}$, $\dot{\gamma}$;
  (*ii*) when needed, requiring a given normal load factor e.g. turn $\Rightarrow \phi_{\mathrm{W}}$, or pull-up/push-over $\Rightarrow q$;
  (*iii*) and possibly assigning aircraft control e.g. aileron failure, $\delta_a = \delta_{a,\mathrm{fail}}$;
  (*iv*) constraining the values of some other state variables according to the kinematic equations (e.g. deriving the values of p, q, r); and
  (*v*) finding the minimum of J as a function of the remaining non-frozen variables.

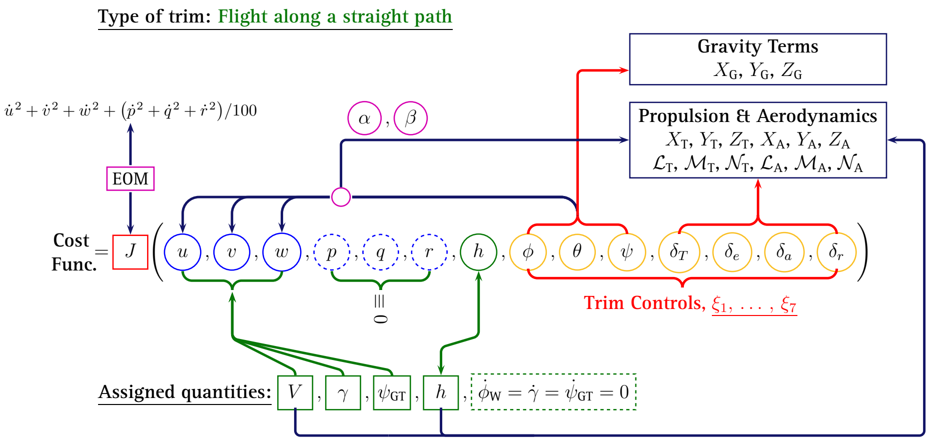
Example: trim for straight flight.



By "straight flight" we mean: flight along a straight path, whatever the bank ϕ, whatever the difference $\psi - \psi_{GT}$

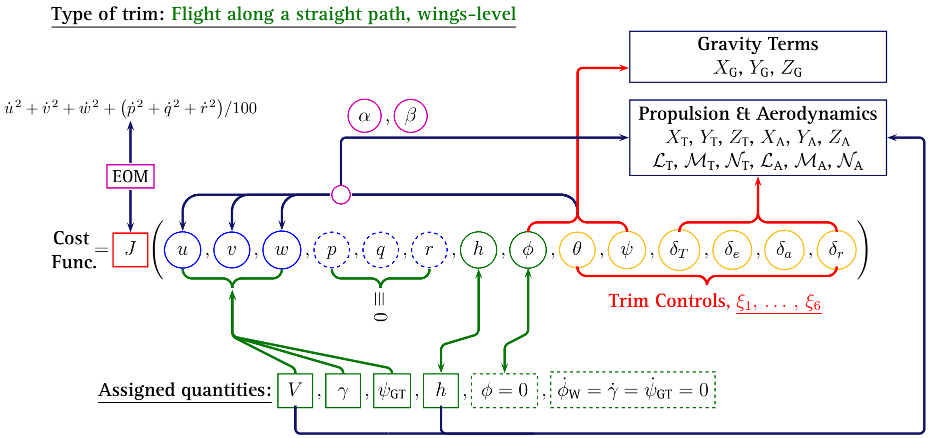
Cost function's independent variables.

Straight flight. Trim controls $\xi = \{\xi_1, \ldots, \xi_7\}$



Cost function's independent variables.

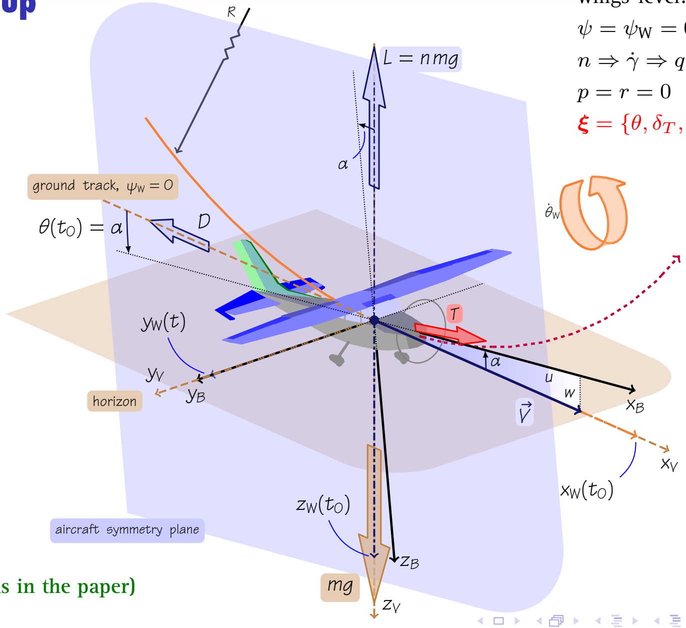
Straight, wings-level flight. Trim controls $\xi = \{\xi_1, \ldots, \xi_6\}$



Numerical trim procedure. Minimization of J.

- The combination $\{\xi_1,\ \xi_2\ \ldots\}_{\text{eq}} = \boldsymbol{\xi}_{\text{eq}}$ for the desired trim condition is a minimum of the multivariate cost function $J(\boldsymbol{\xi})$.
- Symmetric aircraft models and linear aerodynamics included as particular cases.
- The cost function may have many possible minima: many possible $\boldsymbol{\xi}_{\text{eq}}$.
- The generic flight dynamics model JSBSim served as a workhorse to prove our trim strategy: numerical minimization by direct search methods.
- Aim: the development of a practical tool to trim a generic aircraft model in 6DOF. A building block for the linearization of an high fidelity airplane aerodynamic model.

Pull-up



wings-level: $\phi = 0$

$\psi = \psi_W = 0$

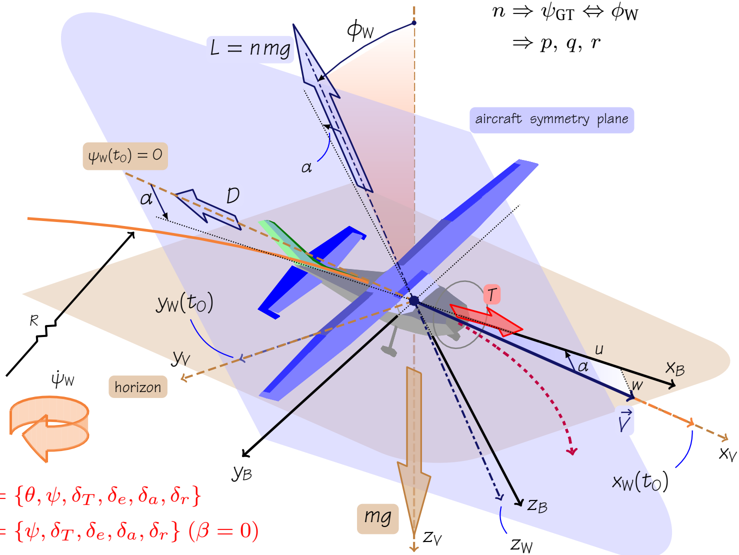
$n \Rightarrow \dot{\gamma} \Rightarrow q$

$p = r = 0$

$\boldsymbol{\xi} = \{\theta, \delta_T, \delta_e, \delta_a, \delta_r\}$

(details in the paper)

Turn



$\boldsymbol{\xi} = \{\theta, \psi, \delta_T, \delta_e, \delta_a, \delta_r\}$

$\boldsymbol{\xi} = \{\psi, \delta_T, \delta_e, \delta_a, \delta_r\}\ (\beta = 0)$

(details in the paper)

Outline

JSBSim: an open source, platform-independent, flight dynamics model in C++

- JSBSim has been under development for 10+ years.
- It runs on Windows, Mac, Linux, IRIX, etc.
- All source code is available. (Perfect example of *Object-Oriented Programming*)
- Open Source tools are all that is needed to build and use it.
- It is scriptable.
- It can be run from a stub application (in *standalone mode*) or integrated within a larger application framework such as OpenEaagles, or a simulation such as FlightGear.
- Philosophy: *to find a balance between fidelity and simplicity, so the task of simulating the flight of any aerospace vehicle can be done with the minimum specific input possible (XML-based).*
- Target audience: upper level engineering students.
- JSBSim also provides advanced capabilities for those who need it.
- `http://www.jsbsim.org`

Trim capability in JSBSim

```
// the sim executive
JSBSim::FGFDMExec* fdmExec = new JSBSim::FGFDMExec();
// loading A/C data
fdmExec->LoadModel(aircraftPath, enginePath, aircraftName);
// initial conditions cfg file
JSBSim::FGInitialCondition *ic = fdmExec->GetIC();
ic->Load(initFileName);

// the trim object
JSBSim::FGTrimAnalysis
    fgta(fdmExec, (JSBSim::TrimAnalysisMode)1);
    // 0: Longitudinal, 1: Full, 2: Full,Wings-Level,
    // 3: Coordinated Turn, 4: Turn, 5: Pull-up/Push-over
    // Low level coding of new trim targets possible

// the ic cfg file contains trim directives
fgta.Load(initFileName);
// optimize J
if ( !fgta.DoTrim() )
    cout << "Trim Failed" << endl;
fgta.Report();
```

Typical JSBSim initialization file

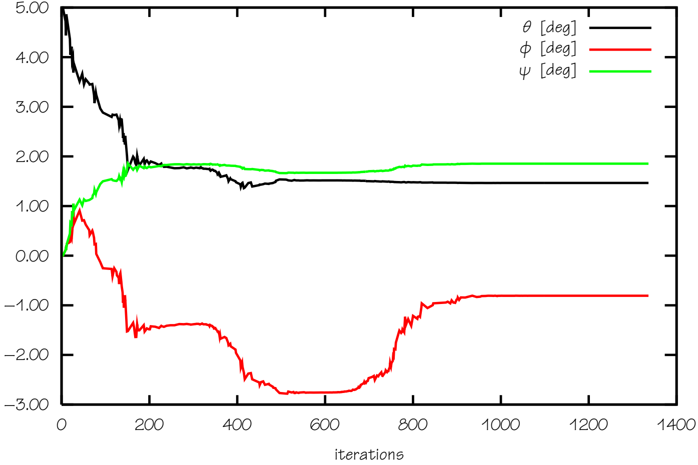
```
<?xml version="1.0"?>
<initialize name="init_for_full_trim">
    <latitude  unit="DEG">    0.0  </latitude>
    <longitude unit="DEG">   0.0  </longitude>
    <altitude  unit="FT">  5500.0  </altitude>
    <psi       unit="DEG">    0.0  </psi>
    <vc        unit="KTS">   88.0  </vc>
    <gamma     unit="DEG">    0.0  </gamma>
    <theta     unit="DEG">    5.0  </theta>
    <alpha     unit="DEG">    5.0  </alpha>
    <phi       unit="DEG">    0.0  </phi>
    <running value="1"/>
                                          <!-- Now set up the trim parameters -->
    <trim_config name="trim01" type="FULL">
        <search type="Nelder-Mead"/>
        <initial_values>
            <phi action="From-IC"/> <theta action="From-IC"/> <psi action="From-IC"/>
            <throttle_cmd>  0.9  </throttle_cmd>
            <elevator_cmd> -0.1  </elevator_cmd>
            <rudder_cmd>    0.0  </rudder_cmd>
            <aileron_cmd>  -0.2  </aileron_cmd>
        </initial_values>
        <steps>
            <phi   unit="DEG"> 0.3 </phi>
            <theta unit="DEG"> 0.3 </theta>
            <psi   unit="DEG"> 0.3 </psi>
            <throttle_cmd>     0.2 </throttle_cmd>
            <elevator_cmd>     0.2 </elevator_cmd>
            <rudder_cmd>       0.2 </rudder_cmd>
            <aileron_cmd>      0.2 </aileron_cmd>
        </steps>
        <output_file name="trim-log.txt"/>
    </trim_config>
</initialize>
```

Gradient-free minimization: *Direct search*

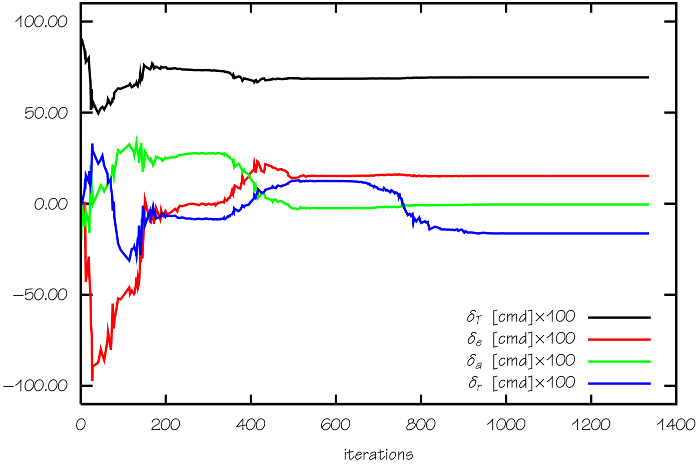
- Direct search methods: optimization techniques that do not explicitly use derivatives (proposed and widely applied in the 1960s).
- Very much used in simulation-based optimization: Complex physical systems designed by optimizing the results of computer simulations (When gradient-based methods are difficult to apply).
- A comprehensive review of direct search methods:
  Kolda, T. G.; Lewis, R. M.; Torczon, V.:
  Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. SIAM Review, 45 (2003), pp. 385-482.
- We have incorporated into JSBSim the *DirectSearch* C++ library:
  Torczon, V. et al.: C++ **DirectSearch** Classes.
  Software available at
  `http://www.cs.wm.edu/~va/software/DirectSearch/`
- We have used the classical Simplex method of Nelder & Mead.
  Treatment of trim control bounds is based on a penalty approach.

Outline

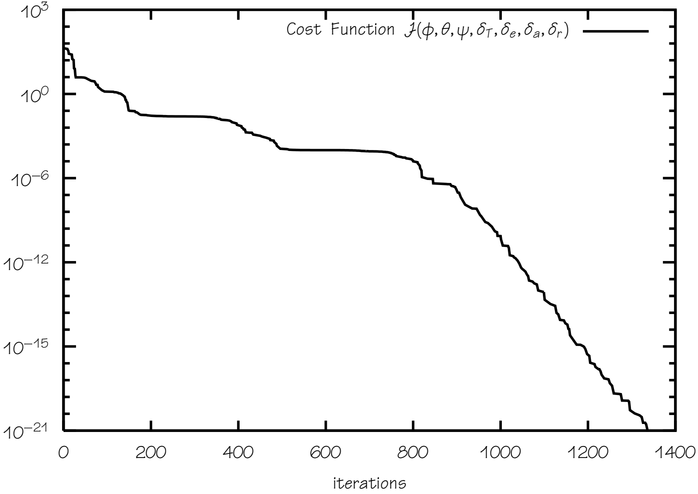
Straight flight. Full trim.
Convergence histories: Euler angles.



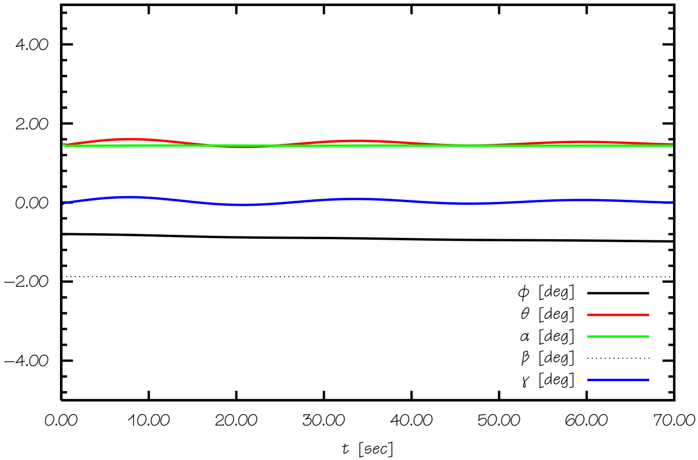
Straight flight. Full trim.
Convergence histories: commands.



Straight flight. Full trim.
Convergence histories: cost function.

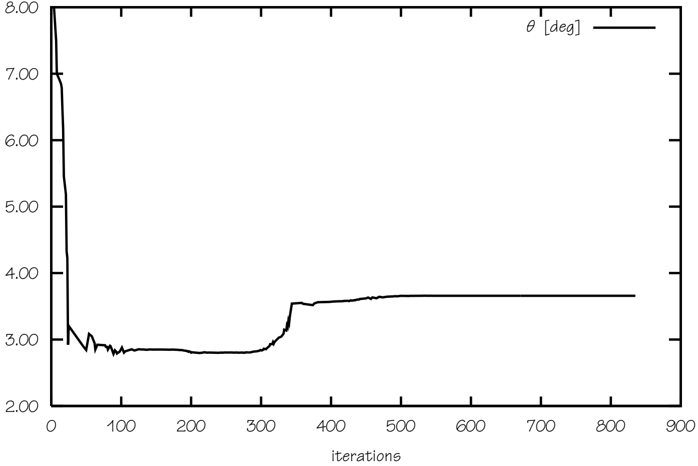


Example of simulation run starting from previous results.

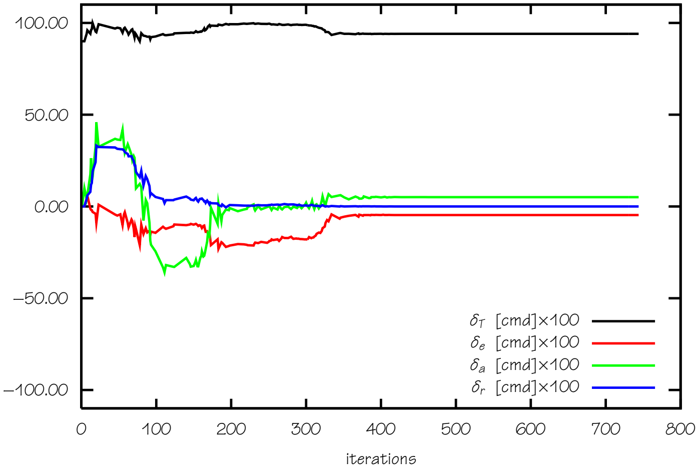


Outline

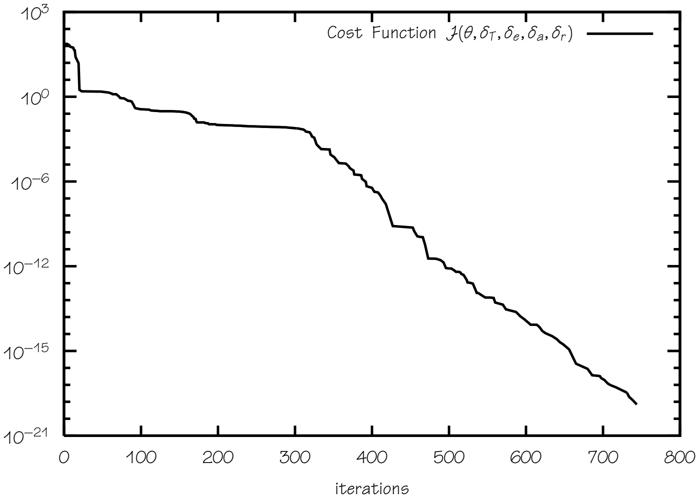
Pull-up, wings-level.
Convergence histories: Euler angles.



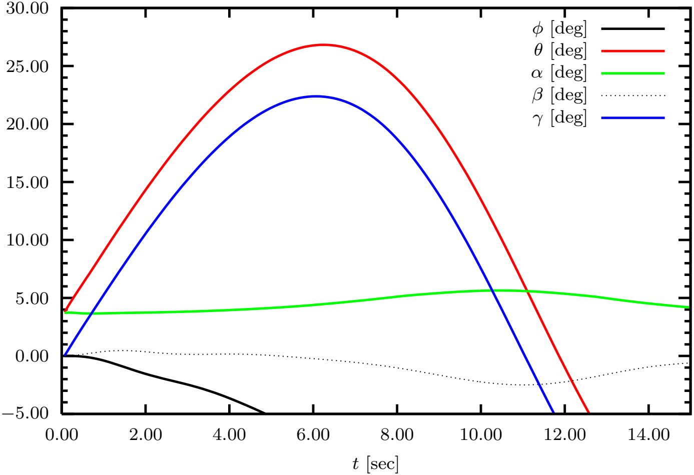
Pull-up, wings-level.
Convergence histories: commands.



Pull-up, wings-level.
Convergence histories: cost function.



Example of simulation run starting from previous results.



Summary

- A general approach to the trim problem (up-and-away flight).
- Trim algorithm based on the constrained minimization of a cost function. Gradient-free numerical minimization method: DirectSearch.
- Examples of trim conditions and trim results has been shown.

- Outlook
  - Some more trim conditions: spin, barrel-roll, asymmetric thrust. On-ground trim.
  - Try the `Opt++` library by *Sandia Labs* (LGPL). (Direct search, numerical gradients)