Flight Dynamics in Microsoft Flight Simulator

An Analysis of the Aircraft Specific Input Parameters of FS 2004 and FSX

Version 1.0

July 2012

By Yves Guillaume

fsfan@intergga.ch

Content

1 Introduction

1.1 Abstract

Overview

This document is an attempt to give a deeper view at the flight dynamics model incorporated in Microsoft Flight Simulator 2004 and FSX. It mainly analyses the interactions between the parameters in the airplane configuration files, aircraft.cfg and *.air, with focus on forces (lift, drag, side force) and turning moments (pitch, roll, yaw).

All formulas are based on flight testing in MSFS and compared with traditional literature of flight models.

Scope

This version focuses on the interactions of aerodynamic forces and moments including propulsion effects. All the findings are based on many hours of flight testing in FS2004 and FSX, supported through great tools and sources mentioned at the end of this document. Modified aircraft.cfg and AIR-files were used to isolate the effects. The findings are all the results of trial & error and might not be complete or fully correct. But in general they showed good consistency with common theory.

This version does not provide specific values for stability derivatives. The estimation of those parameters can be determined by methods covered in several books and internet resources (e.g. DATCOM). At the end of this document there are some links to real aircraft stability derivatives.

Furthermore, the engine modeling and gear contact points are also out of scope of this version.

Target audience

This document is intended for flight dynamics designers of free- and payware products for MSFS and P3D or just anyone who is interested in this topic. The goal is to share know-how to improve the aircraft specific flight dynamics released by the designers.

Personal note

I decided to write this document because I was not satisfied with the information provided in the MSFS SDK related to the aircraft.cfg and AIR-file parameters. So I collected and restructured my personal notes that became spread over Excel and Word documents over several years. I re-tested all the findings again in FS2004 and FSX which led to many new insights about the MSFS flight model.

I first started examining the flight model in MSFS 95 and got a better insight with every FS version released. There were no basic changes in the flight model over all these years and from FS2004 to P3D only some secondary parameters were added.

Now that MS Flight has been released, the traditional aircraft.cfg and AIR-File are gone and the parameters are packed in a proprietary file format protected against editing. For this reason MS Flight is not covered here. But it seems that FSX and P3D will remain popular for many more years.

I do not work in the aviation industry and I have not attended a course in aerodynamics or similar, it was all self-study. I am not in contact with MS or LM.
I am Swiss and my mother tongue is German, but the English spell checker in Word 2010 did a good job!

1.2 Review of the Flight Model

When judging the flight model incorporated in MSFS one can conclude that it is of an advanced degree and much more than a ‘game’. The calculation of the aircraft motion is based on the same formulas found in common textbooks. And most of the relevant aircraft parameters can be set in the AIR-file and aircraft.cfg.

However, the focus of the simulation is on the normal flight envelope. At very high angles of attack (AoA) or very big side slip angles the flight model is not fully accurate. Although the AIR-file includes many control parameters (lookup tables) as a function of AoA (but none of side slip angle Beta), there are limitations especially for aerobatic flights. But these issues are also found in real commercial simulators where flight data outside the normal operation is often not available or only estimated.

In general a detailed modeling of a specific aircraft can be achieved, provided that the input parameters are accurate. Unfortunately, stability derivative data for specific planes are usually not made available for the public by the manufacturers. Judged by the AIR-files of the default planes, even Microsoft seems not have gotten full data sets of Boeing, Cessna etc.
However, there are many studies accessible on the Internet (e.g. NASA/NACA archive) that include data of wind tunnel or flight tests for specific aircraft types. This data is hardly ever complete and then needs adaptation to the MSFS flight model.
The other option is to use formulas for stability derivatives based on the shape of a plane. These methods are not exact, but deliver a sufficient approximation.

The major drawbacks of the aerodynamic flight model from a personal perspective are:

- **Stall angle:** Only one stall angle of attack can be defined per aircraft which is used for all flap settings and Mach numbers. In the real world stall AoA varies significantly with flaps and Mach number. This means that stall speeds and stall behavior in MSFS are only accurate for the designed situation (e.g. full flaps, low Mach).
- **Flaps:** In the real world extended flaps influence most of the stability derivatives noticeably. In MSFS only lift, drag and the pitch moment are affected.
- **Autopilot:** The autopilot is too sensitive mainly in the pitch axis at high altitude and speeds (oscillation) when real stability derivative data is used. This might be explained by the fact that the MS default jets have a much too high pitch damping (mainly Cm_q) and the autopilot was adjusted for these settings. Unfortunately, the major autopilot settings are hard coded in the sim1.dll and cannot be tuned aircraft specific. In the aircraft.cfg only a few autopilot limits can be set. In earlier FS versions the autopilot PIDs could be controlled by R1199 in the AIR-file.
- **Drag:** The drag calculation is simplified and prevents an accurate modeling over the full normal envelope. It can be exactly tuned for a certain flight level and Mach number, but it will not match in other situations (intermediate cruise levels or speed).
- **Mach tables:** The Mach tables in R4xx are fixed steps of M 0.2 and do not allow fine-tuning at intermediate Mach numbers (e.g. M 0.85). This prevents modeling of some typical Mach effects at high cruising speed.
- **Yaw Damper:** The yaw damper does not a good job and heavy planes such as the B747 will experience some yawing oscillation after a turn. Tweaking of some yaw and roll moment coefficients can improve the yawing behavior.

1.3 Simulation Versions

The findings in this document are based on flight tests in FS2004 (FS9), FSX SP1 and SP2. They should apply to FSX Acceleration Pack as well except where noted.
At the time of this release Prepar3D (P3D) version 1.3 is current. As far as I know the flight model is still the same as in FSX, but since I do not own P3D yet, the findings were not verified.

All SDK from FS2004 to P3D 1.3 were used as information sources.

This document does not apply to MS Flight.

1.4 Notations and Glossary

General notations	
Units	The units used in formulas are the same as found in the AIR-file and aircraft.cfg (in general US units are used) to avoid conversions.
Degree vs. radian	MSFS uses both rad and degrees in the AIR-file and degrees only in the aircraft.cfg. Since stability derivatives in the AIR-file are in rad this documentation also uses rad for angles such as AoA, Beta and deflection angles. Conversions from degree to rad are indicated where necessary. 1 rad = 180/pi = 57.29578 deg
Stability derivative e.g. $\frac{\partial R1539\_CL_{\delta f}}{\partial \delta f}$	This is the expression used for stability derivatives as found in the AIR-file. It is the gradient per rad i.e. the first derivative of a linear formula. R stands for **r**ecord.
Aerodynamic coefficient e.g. $CL_{\delta f}$	This is the expression used for coefficients at a specific flight situation (e.g. AoA, Beta, flaps deflection). It is normally the result of a stability derivative multiplied by the corresponding parameters.
Aircraft.cfg parameters e.g. wing_area	Parameter names from aircraft.cfg are used 1 to 1.
General glossary	
AC	Wing aerodynamic center, defined in R1534.
CG	Center of gravity.
$\Delta CG\ lon_{AC}$	Center of gravity longitudinal offset from wing aerodynamic center, ft., positive=forward.
$\Delta CG\ lon_{VMO}$	Center of gravity longitudinal offset from visual model origin, ft., positive=forward.
$\Delta CG\ lat_{VMO}$	Center of gravity lateral offset from visual model origin, ft., positive=right.
$\Delta CG\ vert_{VMO}$	Center of gravity vertical offset from visual model origin, ft., positive=up.
D	Propeller diameter, ft.
M	Mach number.
MAC	Mean aerodynamic chord, ft.
MDL	Visual model of the aircraft, file extension .mdl.
MGC	Mean geometric chord, ft. = wing area/wing span.
$\bar{q}$	Dynamic pressure, lbs/ft^2, ½*rho*V^2 .
$\bar{q}_{propwash}$	Additional dynamic pressure induced by the propellers, ft./s.
SD	Stability derivative.
SDK	Software Development Kit.
V	Airplane true airspeed along flight path, ft./s.
ΔV_n	Propeller induced additional airspeed (TAS), ft./s.
VMO	Visual model origin. The reference point set by the designer of the visual airplane model. It is the starting point for all other reference offsets.

1.5 MSFS Sign Conventions

The sign conventions in MSFS for axes, angles, force and moment coefficients are partly reversed compared to the ones found in most books.
This document uses mainly the same conventions as in the AIR-file and aircraft.cfg.
However, for the summation of total forces (F_X, F_Y, F_Z) and moments (L, M, N) the traditional sign convention is applied to be consistent with common theory.

The following table shows the sign convention in this document. Deviations from the traditional system are indicated in the comment column.

Name	Abbr.	Positive sign	Negative sign	Comment
Airplane Axes				
Longitudinal	X	Forward	Backward	
Lateral	Y	Right	Left	
Vertical	Z	Downward	Upward	MSFS: Positive = upward
Forces				
Lift coefficient	CL	Upward force	Downward force	Reference axis: See chapter 5.3.1
Drag coefficient	CD	Backward force	Forward force	Reference axis: See chapter 5.3.1
Side force coefficient	CY	Force to the right	Force to the left	Reference axis: See chapter 5.3.1
Thrust, lbs	T	Forward thrust	Backward thrust	Acts along the thrust line (may be offset from body axis)
Longitudinal Force (body-fixed), lbs	F_X	Forward force	Backward force	Acts along X-body-axis of the airplane
Lateral Force (body-fixed), lbs	F_Y	Force to the right	Force to the left	Acts along Y-body-axis of the airplane
Vertical Force (body-fixed), lbs	F_Z	Downward Force	Upward Force	Acts along Z-body-axis of the airplane
Moments				
Pitch moment coefficient	Cm	Pitching down	Pitching up	Traditional theory has sign reversed
Roll moment coefficient	Cl	Rolling left	Rolling right	Traditional theory has sign reversed
Yaw moment coefficient	Cn	Yawing right	Yawing left	
Pitch moment, lbs-ft.	M	Pitching up	Pitching down	
Roll Moment, lbs-ft.	L	Rolling right	Rolling left	
Yaw Moment, lbs-ft.	N	Yawing right	Yawing left	
Body Rates				
Pitch rate, rad/s	q	Pitching up	Pitching down	Traditional theory has sign reversed
Roll rate, rad/s	p	Rolling left	Rolling right	Traditional theory has sign reversed
Yaw rate, rad/s	r	Yawing right	Yawing left	
Body angles				
Angle of attack (AoA) (alpha), rad	α	Nose above flight path	Nose below flight path	AoA in MSFS is always with reference to the fuselage X-axis defined in the MDL, not to the wing.
Sideslip angle (beta), rad	β	Nose heading left of flight path	Nose heading right of flight path	
Bank angle (phi), rad	ϕ	Right bank	Left bank	
Pitch angle (theta), rad	θ	Nose up	Nose down	
Surface control angles				
Elevator deflection, rad	δ_e	Elevator up	Elevator down	Traditional theory has sign reversed
Lift spoiler deflection, rad	δ_s	Lift spoiler up (downward force)	N/A	

Name	Abbr.	Positive sign	Negative sign	Comment
Aileron (left) deflection, rad	δ_a $\delta_{a\ left}$	Aileron down (rolling right)	Aileron up (rolling left)	Reference aileron in MSFS
Aileron (right) deflection, rad	$\delta_{a\ right}$	Aileron down (rolling left)	Aileron up (rolling right)	
Roll spoiler (left) deflection, rad	$\delta_{s\ left}$	Roll spoiler up (rolling left)	N/A	The sign is reversed compared to ailerons
Roll spoiler (right) deflection, rad	$\delta_{s\ right}$	N/A	Roll spoiler up (rolling right)	See above
Rudder deflection, rad	δ_r	Rudder right (right yawing)	Rudder left (left yawing)	Traditional theory has sign reversed
Stabilizer deflection (=elevator trim), rad	δ_{etr}	Stabilizer leading edge down (=nose up pitch)	Stabilizer leading edge up (=nose down pitch)	Elevator does not move
Aileron trim deflection, rad	δ_{atr}	Rolling right	Rolling left	Aileron does not move
Rudder trim deflection, rad	δ_{rtr}	Yawing right	Yawing left	Rudder does not move

1.6 Priority of AIR-File Tables

The AIR-file includes all non-dimensional parameters directly related to stability derivatives and engines.
In earlier version of MSFS all SD were in R1101 and scaled by 2048. Although those records are still supported, all SD can be set via records 1539-44, too. The advantage of the latter method is the number format of the values (no scaling required). For the flight model it does not matter which records are used. However, it is important to understand the priority logic. R1539 to 1544 have priority over R1101 as described in the following table.

In this document the naming of SD refers to R1539 to 1544, but the formulas also apply for the equivalent parameters in R1101.

If a 'trigger record' exists it will force MSFS to use the 'active records' and to suppress 'ignored records':

Trigger record	Active records	Ignored records
R1539 Lift SD *R1539 is the only one to trigger the block 1539-44. No individual activation of e.g. 1542 is possible. SD in R1101 are either active or ignored ('all or nothing' principle)*	R1539 to 1544	R1101 all SD *FS2004: Spoiler extending time and g-limits remain active regardless of R1539 existence because there is no equivalent parameter in the aircraft.cfg*
R1545 Lift vs. AoA	R1545 Lift vs. AoA	R404 Lift vs. AoA
R1546 Pitch mom vs. AoA	R1546 Pitch mom vs. AoA	R473 Pitch mom vs. AoA
R1547 Wing aerodynamic center *FS2000/2 only*	R1547 Wing aerodynamic center *FS2000/2 only*	R1534 Wing aerodynamic center *FS2000/2 only. In FS2004/FSX R1547 is never read*

2 Reference Points

2.1 Overview

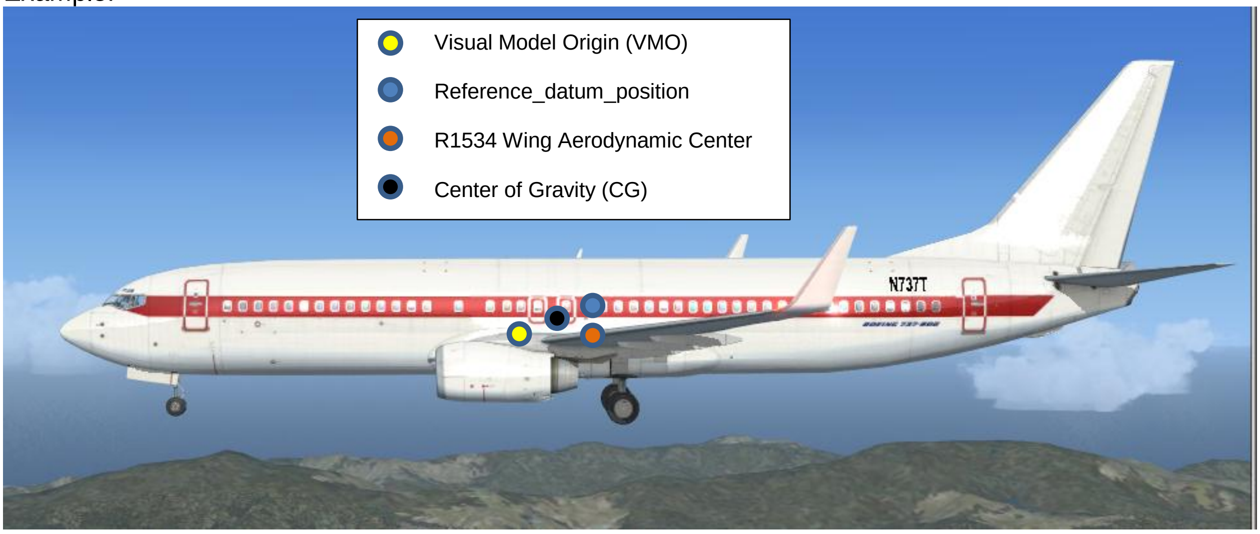
MSFS works with three fixed aircraft reference points in order to calculate flight dynamics and to display some visual features such as lights. The fourth point is the Center of Gravity (CG) and is dynamically calculated in-flight.

- **1. Visual Model Origin (VMO)** as defined in the MDL-File by the designer and affecting
  - Pitch moments (in conjunction with R1534)
  - Roll moments
  - Yaw moments

Three offset points from the VMO:

- **2. *Reference_datum_position*** in aircraft.cfg affecting
  - Weight and fuel distribution (CG position)
  - Moments of Inertia
  - Contact points
  - Engine location
  - Lights
  - …
- **3. Wing Aerodynamic Center**
  - Set via R1534 (if existing) or
  - set by MSFS internally (if R1534 missing)
  - Only affects pitch moments (but not roll and yaw moments)
- **4. Center of Gravity (CG)**
  - Calculated based on empty weight, current payload and current fuel load

Example:



2.2 Visual Model Origin (VMO)

The starting point for all offsets is the visual model origin specified by the designer of the visual model. It is the visual rotation point and designers should set it close to a typical CG position and ahead of the main gear.
The location of the visual model origin is coded in the .mdl-File and cannot be changed without having the source files for the visual model.

2.3 Reference_datum_position

The reference point for the parameters in aircraft.cfg is defined by *reference_datum_position* under [weight_and_balance]. It is the offset in feet from the visual model origin (VMO) and affects parameters in aircraft.cfg such as CG, contact points, airplane geometry etc.

The location of this point can freely be chosen. It could be at the nose of the plane like some MS default aircraft or close to the visual model origin.

When gear contact points are defined, it is important to set the correct longitudinal offset from the *reference_datum_position* and therefore CG position. The distance between the main gear and the CG has a very noticeable effect at take-off rotation.

Recommendation for *reference_datum_position*:
It is helpful to use the same offset position as in R1534 in the AIR-file to avoid dealing with two different reference points

- Longitudinal position: 25% MAC (typical position for nominal SD)
- Lateral position: Center of fuselage (i.e. zero offset)
- Vertical position: Fuselage reference line (about in the middle of the fuselage height)

The offsets can approximately be found by looking at the visual model from spot view in MSFS.

2.4 Wing Aerodynamic Center (R1534)

The SD moments in the AIR-file are nominal values and by concept apply at one specific CG position. For pitch moments (Cm) this location can be explicitly set in R1534 if it differs from the visual model origin.
If R1534 is omitted in the AIR-file, MSFS tries to calculate an 'appropriate' offset itself based on parameters in the AIR-file and aircraft.cfg. Both ways are described in the following chapters.

2.4.1 AIR-File with existing R1534 Wing Aerodynamic Center

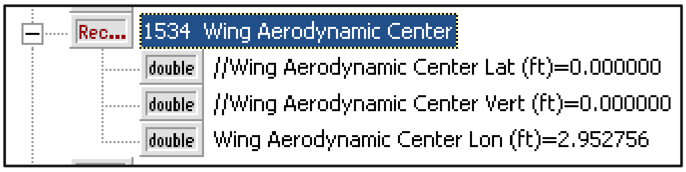
The AIR-file can optionally include R1534 which is an offset from the VMO. In FS2004 and FSX only the longitudinal offset parameter (R1534_3) works and only affects pitch moments. The lateral and vertical offsets have been disabled (=0) after FS2002.

The nominal pitch moments values in the AIR-file are therefore valid for the CG position defined in R1534_3. This location is typically chosen at 25% MAC in literature which is close to the aerodynamic center of the main wing at low speed. However, the 25% MAC position is just an arbitrary position. Any other offset could be chosen as long as the nominal pitch moments in R1541, R473/1546 and R42x (Mach tables) of the AIR-File correspond to this location.

In other words, when the CG in MSFS is exactly at the longitudinal offset defined in R1534_3, the nominal pitch SD values apply (e.g. $\frac{\partial Cm}{\partial \alpha}$ in R473/1546). If the CG shifts away (e.g. by fuel flow) the effective pitch moment in-game will differ from the nominal ones in the AIR-file, meaning their effective gradients become steeper or more flat.

Note that R1534 refers to the wing aerodynamic center, not to the one of the complete aircraft (at around 40-50% MAC). The latter is indirectly defined by total dCm/dCL relationship and is known as neutral point.

Nominal roll and yaw moments always apply for a CG position at VMO.



Recommendation for R1534 Wing Aerodynamic Center:
I recommend including R1534 in the AIR-file because if it is omitted, MSFS calculates the SD reference point itself and controlling it becomes more difficult (see Appendix 1):

- **R1534 Lateral offset (ft):** Disabled in FS2004 and FSX. Any values are ignored and assumed 0. This is not critical since there is no lateral offset from the visual model origin. Positive = Wing aerodynamic center right of visual model origin.
- **R1534 Vertical offset (ft):** Disabled in FS2004 and FSX. Any values are ignored and assumed 0. This is not critical since the vertical offset from the visual model origin is normally small. Positive = Wing aerodynamic center above visual model origin.
- **R1534 Longitudinal offset (ft):** Affects pitch moments only in FS2004 and FSX, not roll and yaw moments. Positive = Wing aerodynamic center ahead of visual model origin
  - For consistency reasons set it to the same offset value as in *reference_datum_position* to have only one reference point for both files. I recommend using an offset that corresponds to the 25% MAC position.
  - The reference for yaw and roll moments is always the visual model origin. Any offset values are treated as zero. This means that if we have real world data for e.g. $\frac{\partial Cn}{\partial \beta}$ at 25% MAC but the visual model origin is at 60% MAC, a correction to the known SD value at 25% should be applied in the AIR-file. In this example this gives:

$$\frac{\partial Cn\_at\ 60\%\ MAC}{\partial \beta} = \frac{\partial Cn\_at\ 25\%\ MAC}{\partial \beta} + \frac{\partial CY}{\partial \beta} * (0.6 - 0.25) * \frac{MAC}{wing\_span}$$

R1534 is a legacy parameter partially kept alive for backward compatibility. Introduced in FS2000 (Concorde), it consisted of a working lateral, vertical and longitudinal offset parameter and influenced pitch, roll and yaw moments. It was basically the equivalent to *reference_datum_position* just for AIR-files.

In FS2002 the record existed in all MS standard planes. FSEdit 2002 inserted R1547 which had the same function and was overriding R1534. FSEdit2002 computed the longitudinal offset in R1547 through dividing the linear gradient of $\frac{\partial Cm}{\partial \propto}$ (R473/1546) by $\frac{\partial CL}{\partial \propto}$ (R404/1545) and multiplying the result with MAC. If R1534 or 1547 were omitted in FS2002 all three parameters were assumed to be zero.

In FS2004 MS removed all R1534 sections in their default aircrafts (except the Wright Flyer and the Schweizer sailplane). The longitudinal offset parameter in R1534 remained the only working parameter but did only affect pitch moments. R1547 was also deactivated and FSEdit 2004 did not add it anymore. *Wing_incidence* and *wing_twist* in the aircraft.cfg became inactive, too. This led to a slightly different flight behavior of old FS2000/02 aircrafts when used in FS2004 and FSX.

2.4.2 AIR-File without existing R1534 Wing Aerodynamic Center

If R1534 does not exist in an AIR-file in FS2004 and FSX, MSFS calculates the wing aerodynamic center (i.e. the reference point for pitch moment SD) in its own way. As a result the SD longitudinal reference point for pitch moments is always exactly in the middle of the zero fuel weight CG position and the aerodynamic center of the plane (not wing). The exact method used by MSFS is described in Appendix 1.

3 Weight & Balance

3.1 Mean Aerodynamic Chord

3.1.1 Overview

The Mean Aerodynamic Chord (MAC) of the main wing is used for quantifying pitch moment coefficients but also for indication of the CG position. MAC length and MAC position is defined by the airplane manufacturer and often published in the FAA type certification data sheets.

In MSFS MAC length and MAC position cannot be directly entered but are calculated from other parameters. Additionally, MSFS allows setting a second MAC length which is only used for gauges and can be different from the one computed for aerodynamic purpose.

3.1.2 MAC Length Affecting Flight Dynamics

In MSFS the MAC length used for flight dynamics cannot directly be entered. Instead it is calculated by MSFS based on the following data in aircraft.cfg under [airplane_geometry]:

- *wing_area* (sq. ft.)
- *wing_span* (ft.)
- *wing_root_chord* (ft.)

Wing_area and *wing_span* are fixed values, so *wing_root_chord* is the control parameter for MAC calculation. The labeling *wing_root_chord* is not quite correct. It is actually the body centerline chord of a theoretical linearly tapered wing plan form. *Wing_root_chord* is about equal to MAC if the wing is not tapered. The following formula gives the correct *wing_root_chord* for a desired MAC in MSFS.

$$wing\_root\_chord = MGC + \sqrt{0.25 * (2 * MGC)^2 - 4 * MGC^2 + 3 * MGC * MAC}$$

Where:

MGC = Mean geometric chord length in ft. = wing_area / wing_span

MAC = real world MAC length in ft.

3.1.3 MAC length for gauge indication

Normally, the MAC length used for flight dynamics is the same as the one used for indication of CG position in gauges. However, MSFS provides a second MAC length variable which only affects gauge indications. This parameter is in the AIR-file in R1515 (*wing specs 2*) and alternatively in the aircraft.cfg. It does not have a physical effect on the flight model. But it is useful in the application case described further below.

- AIR-file: **R1515_1** (OBSOLETE_AIR_70_WING_SPECS_2) (first parameter): MAC length (ft).
- Aircraft.cfg: ***wing_cg_ref_chord*** (ft) in [airplane_geometry]: This is a parameter not documented in the SDK.

The reading priority in MSFS is as follows:
1. If *wing_cg_ref_chord* exists in aircraft.cfg, R1515 is ignored.
2. If *wing_cg_ref_chord* does not exist in aircraft.cfg, the value of R1515 is used.
3. If R1515 does not exist too, the same MAC length calculated for flight dynamics is taken for gauge indication.

Since *wing_cg_ref_chord* is an optional parameter, it is normally missing and the obsolete R1515 is read. This might not be desired and <u>it is recommended to delete R1515 completely</u>. All other entries in R1515 are in the aircraft.cfg, so this record is not needed.

<u>Application case: MAC smaller than MGC</u>
In MSFS MAC length used for flight dynamics cannot be smaller than MGC length. This becomes a small issue for some airplanes like the Boeing 747-200 where real world MAC length is 27.33 ft. but MGC length is 28.11 ft. (= 5500 sq. ft. / 195.67 ft.). There is a work around:

- *wing_root_chord* is to be set equal to MGC length.
- In aircraft.cfg in [airplane_geometry] ***wing_cg_ref_chord*** should be added and set to the real world MAC length (in ft.) → this gives correct CG position in gauges.
- All pitch moment SD in the AIR-File should be corrected by a factor of MAC/MGC length (e.g. 0.972 for the 747 example).

3.1.4 MAC position

The placing of MAC leading edge to the correct position along the airplane body is important for a correct CG indication in gauges and is set in *wing_pos_apex_lon*. However, it does not affect flight dynamics. MAC location is also available from FAA type certification data sheets (e.g. 747 Classic: ‘Leading edge of MAC is 1258 inches aft of datum’).
MSFS uses the following parameters to determine MAC leading edge position. It is assumed here that *reference_datum_position lon* is set at the standard 25% MAC (measured from VMO).

- MAC length (ft.) (calculated or from *wing_cg_ref_position* or R1515_1)
- *wing_span* (ft.)
- *wing_sweep* at leading edge (deg)
- *wing_pos_apex_lon* (ft.): leading edge of MAC

$$wing\_pos\_apex\_lon = \left(1 - \frac{\frac{wing\_span}{12} * wing\_root\_chord + \frac{wing\_area}{6}}{\frac{wing\_area}{2}}\right) * \frac{wing\_span}{2} * \tan(wing\_sweep) + 0.25 * MAC$$

3.2 Airplane Mass

The gross mass (called ‘weight’ in aircraft.cfg) of the airplane consists of three components in MSFS. Payload (station_load) and fuel weight can be adjusted in-flight via the MSFS menu. This is self-explanatory.

$$Gross\ mass = empty\_weight + \sum_n (station\_load.n\ mass) + \sum (current\ fuel\ mass\ of\ tanks)$$

3.3 Moments of Inertia

The moments of inertia (MOI) can only be approximately calculated because the weight distribution of a plane is not exactly known. MSFS reads the empty weight MOI values from the aircraft.cfg and attempts to adjust them for payload and fuel weight. Unfortunately, MSFS partly fails to do a reasonable approximation for payload MOI which results in too low MOI values. This becomes noticeable in flight when the plane feels less sluggish than it is supposed to be at a certain weight. A manual correction to the empty weight MOI entered in the aircraft.cfg is therefore required. This chapter describes this in detail.

The general difficulty in MOI computation is the fact that every single weight component and its location must be considered individually. Since there is an infinite number of weight stations this problem could be solved by integration, provided the weight distribution is known. But this is not the case in MSFS. Unlike the CG computation it is not possible to simply take the average of all weight components and their positions: This average position would be at the CG and MOI would always be zero.

The general formulas for MOI are:

$$MOI = \sum (mass * mass\ offset\ from\ CG)^2$$

$$Product\ of\ MOI\ (cross\ coupled) = \sum (mass * mass\ offset\ from\ CG)$$

MSFS uses a simplified approach: First it calculates the CG position for the current in-flight weight configuration (empty weight + payload + fuel weight). This is taken as reference location for MOI calculation. There are three weight categories contributing to MOI:

1. Empty weight
2. Payload weight
3. Fuel weight

1. Empty weight

First MSFS reads the empty weight MOI values for roll, pitch, yaw and cross coupled XZ (roll and yaw axis) found in aircraft.cfg under [weight_and_balance]. The MOI unit is slugs*ft^2 (1 slug=32.2 lbs).

- *empty_weight_roll_MOI*
- *empty_weight_pitch_MOI*
- *empty_weight_yaw_MOI*
- *empty_weight_coupled_MOI*

The values entered are valid for a CG location equal to *empty_weight_cg_position*. Any CG offset from there will change the empty weight MOI in MSFS.

2. Payload

Payload contribution to MOI seems not to be accurate in MSFS and needs manual correction (see below). Instead of summing up all *station_load.n* entries individually, MSFS uses the offset of the average location of all payload stations from the CG position.
This means that placing one payload of e.g. 1000 lbs 30 feet ahead of the CG and another one of 1000 lbs 30 feet behind CG results in zero MOI contribution in MSFS, because the average location is exactly at the CG. In reality both stations' moments would sum up.

As a consequence a high number of *station_load* entries do not help to get better MOI estimations (but they affect weight and CG calculation as expected).

For this reason payload MOI contribution in MSFS is significantly too low. The calculated payload MOI values are typically about only 5% of the real values, depending on the current CG position. This can be validated by checking the Simulation variables from FSX SDK:

- TOTAL WEIGHT PITCH MOI
- TOTAL WEIGHT ROLL MOI
- TOTAL WEIGHT YAW MOI
- TOTAL WEIGHT CROSS COUPLED MOI

Work around:
As a work around, the real world payload MOI values should be incorporated in the *empty_weight* MOI values. In other words, the MOI values provided by the designer should be those for a typical zero fuel weight instead of empty weight.

One remaining problem is that changing payload directly via the MSFS menu will not adjust MOI values correctly.

3. Fuel weight

Fuel weight MOI are calculated by MSFS based on the tanks defined in [fuel]. Every tank's current weight and position is individually taken into account. As the tanks become emptier the MOI values in flight will decrease. This is an acceptable approximation.

Summary of MSFS MOI calculation

$$
\begin{aligned}
Total\ pitch\ MOI = {} & empty\_weight\_pitch\_MOI \\
& + empty\_weight/32.2 * ((empty\_weight\_CG\_position_{lon} - CG_{lon})^2 \\
& + (empty\_weight\_CG\_position_{vert} - CG_{vert})^2) \\
& + total\ payload\ weight/32.2 * \ ((total\ payload\ position_{lon} - CG_{lon})^2 \\
& + (total\ payload\ position_{vert} - CG_{vert})^2) \\
& + \sum_{n=0} (fuel\ weight\ tank_n/32.2 * ((fuel\ tank_n\ position_{lon} - CG_{lon})^2 \\
& + (fuel\ tank_n\ position_{vert} - CG_{vert})^2))
\end{aligned}
$$

$$
\begin{aligned}
Total\ roll\ MOI = {} & empty\_weight\_roll\_MOI \\
& + empty\_weight/32.2 * ((empty\_weight\_CG\_position_{lat} - CG_{lat})^2 \\
& + (empty\_weight\_CG\_position_{vert} - CG_{vert})^2) \\
& + total\ payload\ weight/32.2 *\ ((total\ payload\ position_{lat} - CG_{lat})^2 \\
& + (total\ payload\ position_{vert} - CG_{vert})^2) \\
& + \sum_{n=0} (fuel\ weight\ tank_n/32.2 * ((fuel\ tank_n\ position_{lat} - CG_{lat})^2 \\
& + (fuel\ tank_n\ position_{vert} - CG_{vert})^2))
\end{aligned}
$$

$$
\begin{aligned}
Total\ yaw\ MOI = {} & empty\_weight\_yaw\_MOI \\
& + empty\_weight/32.2 * ((empty\_weight\_CG\_position_{lon} - CG_{lon})^2 \\
& + (empty\_weight\_CG\_position_{lat} - CG_{lat})^2) \\
& + total\ payload\ weight/32.2 *\ ((total\ payload\ position_{lon} - CG_{lon})^2 \\
& + (total\ payload\ position_{lat} - CG_{lat})^2) \\
& + \sum_{n=0} (fuel\ weight\ tank_n/32.2 * ((fuel\ tank_n\ position_{lon} - CG_{lon})^2 \\
& + (fuel\ tank_n\ position_{lat} - CG_{lat})^2))
\end{aligned}
$$

$$
\begin{aligned}
Total\ coupled\ MOI\ (XZ) = {} & empty\_weight\_coupled\_MOI \\
& + empty\_weight/32.2 * (empty\_weight\_CG\_position_{lon} - CG_{lon}) \\
& * (empty\_weight\_CG\_position_{vert} - CG_{vert}) \\
& + total\ payload\ weight/32.2 *\ (total\ payload\ position_{lon} - CG_{lon}) \\
& * (total\ payload\ position_{vert} - CG_{vert}) \\
& + \sum_{n=0} (fuel\ weight\ tank_n/32.2 * (fuel\ tank_n\ position_{lon} - CG_{lon}) \\
& * (fuel\ tank_n\ position_{vert} - CG_{vert}))
\end{aligned}
$$

Where

$CG_{lon\ or\ lat\ or\ vert}$	Current CG position offset (feet) from *reference_datum_position*
$fuel\ weight\ tank_n$	Current fuel weight per tank (n) set in MSFS via menu (load editor) in lbs
$fuel\ tank_n\ position_{lon\ or\ lat\ or\ vert}$	Tank position defined in aircraft.cfg in feet, e.g. Center1= **21.98, 0.00, -9.84**, 12891, 32
$total\ payload\ weight$	Sum of the <u>current</u> weight of all payload stations set via MSFS menu (may deviate from the standard values defined in aircraft.cfg)
$total\ payload\ position_{lon\ or\ lat\ or\ vert}$	Average position of all current payload stations set via MSFS menu from *reference_datum_position*.

4 Control surfaces

Only elevator, pitch trim and flaps are described in this chapter because they require some additional explanation. The formulas for all other surface controls can be found directly in the SD provided in chapter 5 and 6.

4.1 Elevator and Pitch Trim

The main pitch control system in MSFS consists of an elevator and a stabilizer trim surface. The pitch trim concept is basically more the one of a moving horizontal stabilizer than that of an elevator with trim tabs. However, the additional lift and drag generated by the deflected stabilizer is not simulated in MSFS. Furthermore, the default autopilot only uses the stabilizer for trim and not the elevator. For consistency with the SDK the term Cm_detr is used although the elevator is not directly involved in trim.

MS seems to have pitch controls designed for joysticks where the stick returns to a fixed center when hand force is released (such as most game sticks, e.g. Saitek X52). Zero hand force means zero elevator deflection in relation to stabilizer, so the stabilizer is the only trim surface.

There are three components related to elevator and stabilizer trim: Stabilizer, elevator and an additional elevator effect due to stabilizer deflection.

$$Cm_{\delta etr+\delta e} = Cm_{\delta etr} + Cm_{\delta e} + Cm_{\delta e(etr)}$$

1. Stabilizer pitch moment

$$Cm_{\delta etr} = \frac{\partial R1541\_Cm_{\delta etr}}{\partial etr} * \delta etr\ [elevator\_trim\_limit] * R536\_Elev\_pitch\_eff\_vs\_AoA$$
$$* R1525\_Elev\_trim\_pitch\_eff\_vs\_\bar{q} * elevator\_trim\_effectiveness$$

This is the pitch moment due to stabilizer trim. Despite its name the R536 scalar also applies to stabilizer not only to elevator.

2a. Elevator pitch moment

$$Cm_{\delta e} = \left(\frac{\partial R1541\_Cm_{\delta e\_basic} + \partial R420\_Cm_{\delta e\_M}}{\partial \delta e}\right)$$
$$* \delta e\ [elevator\_up\ OR\ down\_limit * R517\_Elev\_defl\_limit\_scalar\_vs\_\bar{q}]$$
$$* R341\_Elev\_pitch\_eff\_vs\_deflection * R536\_Elev\_pitch\_eff\_vs\_AoA * elevator\_effectiveness$$

This is the pitch moment due to elevator deflection when stabilizer trim is zero (CG offset pitch effect not pictured here, see chapter 6.3.2).

2b. Elevator pitch moment due to stabilizer deflection

$$Cm_{\delta e(etr)} = \frac{\delta e}{elevator\_up\_limit} * \frac{\partial R1541\_Cm_{\delta etr}}{\partial etr} * \delta etr$$
$$* R341\_Elev\_pitch\_eff\_vs\_deflection * R536\_Elev\_pitch\_eff\_vs\_AoA * elevator\_effectiveness$$

The following description is just my interpretation of the above equation.

This looks to be some kind of a simple stability augmentation system that corrects for different CG positions to attain a more constant (joy-) stick force.
Let’s look at this closer in the real world: A forward CG in general requires nose-up trim mainly due to the steeper slope of Cm_α. This also means that higher elevator deflections are necessary to

reach the same g-load like at an aft CG. In a real plane (e.g. Boeing jets) the pilot's control column hand force then becomes lighter to allow bigger elevator deflections (and control column movements) at still the same hand force. This is the stick force per g concept. The current stabilizer deflection is often used as main reference for hand force adjustments (instead of actual CG location or speed).

In MSFS the resistance force of (non-force-feedback) joysticks cannot be changed of course. Instead the pitch moment commanded per joystick deflection has been tied to the stabilizer deflection in MSFS. When the stabilizer is trimmed nose-up and the joystick is pulled (e.g. at take-off), the resulting elevator pitch up effect (commanded by the joystick) is higher than at a zero trimmed stabilizer.
This means that for example at takeoff the flight sim pilot can use the same joystick deflection (=same hand force) to rotate the plane regardless of longitudinal CG position. Of course this is a rough approximation and does not compensate for different flaps settings etc.

Note that MSFS is a little inconsistent in that the actual elevator deflection per joystick deflection is not changed with CG shift. The additional pitch effect in the real world comes from a higher elevator deflection, but in MSFS the deflection remains the same and the additional pitch effect is 'just there'.

In other words the above equation says that the relation of the current elevator deflection to the defined maximum deflection is used as a scalar on the pitch moment due to stabilizer trim. For example, an elevator deflection of 10° out of 20° will add an extra 50% of the current pitch moment due to stabilizer.

2a and 2b can also be summarized in one equation.

4.2 Flaps

Each flaps set n adds a certain lift, drag and pitch moment according to its individual deflection angle. The basic SD in R1539/40/41 determine the value at 1 rad deflection. The scalars in the [flaps.n] sections of the aircraft.cfg are individual multipliers on each flap set. The formulas are:

$$CL_{\delta f} = \frac{\partial R1539\_CL_{\delta f}}{\partial \delta f} * \sum_{n=0} (\delta f_n * lift\_scalar_n)$$

$$CD_{\delta f} = \frac{\partial R1540\_CD_{\delta f}}{\partial \delta f} * \sum_{n=0} (\delta f_n * drag\_scalar_n)$$

$$Cm_{\delta f} = \frac{\partial R1541\_Cm_{\delta f}}{\partial \delta f} * \sum_{n=0} (\delta f_n * pitch\_scalar_n)$$

Some rules for flaps apply:

<table>
<tr><th>Rules</th><th colspan="2"></th></tr>
<tr><td>Max. number of flaps sets</td><td><u>FS2004</u>
Up to 4 sections
[flaps.0] to [flaps.3]
Numbering can start at 0, 1, 2 or 3 and can be discontinuous</td><td><u>FSX</u>
Up to 6 sections
[flaps.0] to [flaps.5]
Numbering can start at 0, 1, 2, 3, 4, 5 and can be discontinuous
Since the visual model only supports 4 sets, the other 2 can be used to set more lift, drag and pitch effects.</td></tr>
<tr><td>Combination of flaps types</td><td colspan="2">There is any combination allowed between type=1 (trailing edge, TE) and type=2 (leading edge, LE):
e.g.
3 x TE and 1 x LE
or
4 x TE
or
6 x LE
or …</td></tr>
</table>

The visual model supports up to 4 different flaps sets (two for each wing). Those are assigned to a [flaps.n] section by MSFS according to the following rules:

Visual model flaps	**Assignment to [flaps.n] set**
TE inboard	type=1 (TE) set with the lowest [flaps.n] number
TE outboard	type=1 (TE) set with the second to lowest [flaps.n] number
LE inboard	type=2 (LE) set with the lowest [flaps.n] number
LE outboard	type=2 (LE) set with the second to lowest [flaps.n] number

Example of flaps lift calculation:

<u>Input</u>

R1539_CL_df = 2.5

[flaps.0]
Type=1
lift_scalar=0.7
flaps-position.0=0
flaps-position.1=10
flaps-position.2=25
flaps-position.3=40

[flaps.1]
Type=2
lift_scalar=0.5
flaps-position.0=0
flaps-position.1=3 → also applies for flaps-position.2
flaps-position.3=6

<u>Output</u>

At flaps-position.1 the additional flaps lift would be:
CL_df = 2.5 * (10°/57.3 * 0.7 + 3°/57.3 * 0.5) = 0.371

Same method applies for drag and pitch moment.

5 Calculation of Forces

5.1 Summary

MSFS takes aerodynamic forces (lift, drag, side force), propulsion forces (thrust) and weight forces (mg) to calculate the total forces along the three body axes X, Y, Z of the airplane.

Forces along longitudinal body axis (positive=forward):

$$F_{X_{Total}} = F_{X_{Aero}} + F_{X_{Propulsion}} - m * g * \sin(\theta)$$

Forces along lateral body axis (positive=right):

$$F_{Y_{Total}} = F_{Y_{Aero}} + F_{Y_{Propulsion}} + m * g * \cos(\theta) * \sin(\phi)$$

Forces along vertical body axis (positive=downward):

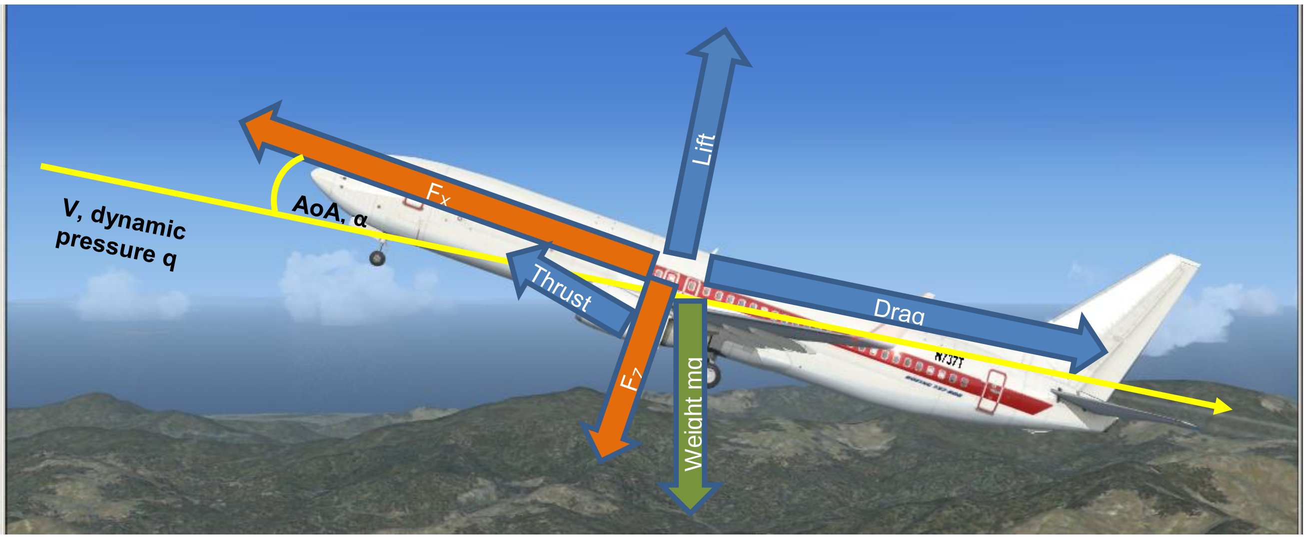
$$F_{Z_{Total}} = F_{Z_{Aero}} + F_{Z_{Propulsion}} + m * g * \cos(\theta) * \cos(\phi)$$

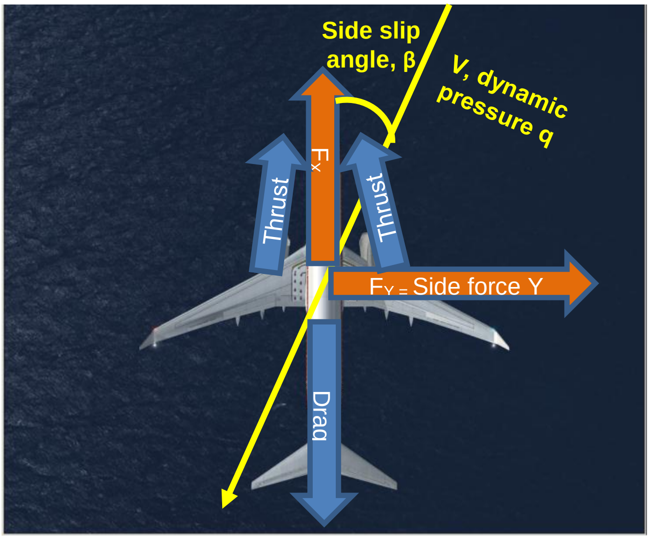
Where:
m gross mass of the airplane (lbs)
g gravity constant, 32.17 slugs/ft2
θ Pitch angle
ϕ Bank angle

These body-fixed forces will then be translated to the earth-fixed coordinates by MSFS (not covered here).

5.2 Axes System

MSFS applies an axis system called ‘stability axes’. This is a mixed approach between body axes and wind axes. Note that the drag acts along the flight path from side view, but along the X-body-axis from top, e.g. it is independent from side slip angle (Beta).





5.3 Aerodynamic Forces

5.3.1 Overview

Aerodynamic forces in this context are lift (L, CL), drag (D, CD) and side force (Y, CY).

The aerodynamic forces will be translated to forces along the body axis (F_X, F_Y, F_Z) at running time of MSFS.

$$F_{X\,Aero} = (CL_{Total} * \sin(\alpha) - CD_{Total} * \cos(\alpha)) * \overline{q} * wing\_area$$

$$F_{Y\,Aero} = CY_{Total} * \overline{q} * wing\_area$$

$$F_{Z\,Aero} = (-CL_{Total} * \cos(\alpha) - CD_{Total} * \sin(\alpha)) * \overline{q} * wing\_area$$

The composition of the individual coefficients with reference to the parameters in the AIR-file and aircraft.cfg is described in detail in the following chapters.

5.3.2 Lift Coefficient

The total lift coefficient CL at a certain flight situation is calculated as follows:

$$CL_{Total} = \left(CL_{\alpha} + CL_{\delta f}\right) * CL\_scalar\_vs\_\alpha * R400\_CL\_ground\_effect\_scalar$$
$$* R401\_CL\_scalar\_vs\_M + CL_{ih} + CL_{\delta e} + CL_{\delta s}$$
$$+ CL_{\alpha dot} + CL_{q}$$

For simplicity the following term can be summarized in CL_α_wf (= wing, flaps)

$$CL_{\alpha\, wf} = \left(CL_{\alpha} + CL_{\delta f}\right) * CL\_scalar\_vs\_\alpha * R400\_CL\_ground\_effect\_scalar * R401\_CL\_scalar\_vs\_M$$

Where:

$$CL_{\alpha} = \frac{\partial R404/1545\_CL_{\alpha\_basic}}{\partial \alpha}[\alpha]$$

This is the basic (clean) lift component for wing-body-tail as a function of the CL vs. AoA curve in R404/1545.

- Note that only one single stall AoA can be defined for an airplane which is done in this table. It is valid for all flaps settings and Mach numbers. Real world stall speeds will therefore not match in most cases, since the real stall AoA varies significantly with flaps/slats and Mach number.

$$CL_{\delta f} = \frac{\partial R1539\_CL_{\delta f}}{\partial \delta f} * \sum_{n=0}(\delta f_n * lift\_scalar_n)$$

This is the lift component due to the individual deflection of each [flaps.n] section in aircraft.cfg and multiplied by the individual *lift_scalar* of each section.

- This value shifts the CL_α_basic curve up (or down).

$R400\_CL\_ground\_effect\_scalar$

This is a lift scalar on the lift curve of the wing-body-tail (R404/1545) plus flaps due to ground effect as defined in R400. It is a function of the relative ground height of the visual model origin.

$$relative\_ground\_height = \frac{ground\ height\ VMO}{wing\_span}$$

$R401\_CL\_scalar\_vs\_M$

This is the lift scalar on the lift curve of the wing-body-tail (R404/1545) plus flaps due to Mach effect as defined in R401.

$$CL\_scalar\_vs\_\alpha\ = cruise\_lift\_scalar * \frac{(\alpha_{CL\_max} - [abs]\alpha)}{\alpha_{CL\_max}}$$

This is a scalar on the lift curve of the wing-body-tail (R404/1545) plus flaps.

- It is a function of the delta between the AoA where CL is maximal in R404/1545 and the current (absolute i.e. positive) AoA.
- *Cruise_lift_scalar* has its nominal value at AoA=0 and decreases linearly with increasing and decreasing AoA. It becomes 1.0 at and above that (positive) AoA where CL is at its maximum (CL_max).
- Along the negative AoA range the scalar acts almost the same way and reaches 1.0 at the same AoA offset from zero (e.g. -15°) as in the positive range. That means for the negative AoA range the lowest (most negative) CL value does not influence the scalar. It is always CL_max that defines both values.
- This scalar can normally be ignored (i.e. set to 1.0) and instead R404/1545 should be modified.

$$CL_{ih} = \left(\frac{\partial R1539\_CL_{ih\_basic} + \partial R413\_CL_{ih\_M}}{\partial ih}\right) * htail\_incidence$$

This is the lift due to the incidence angle of the horizontal stabilizer. *htail_incidence* is a fixed value and does not change in-flight. It is not related to stab trim.

Recommendation:
R1539_CL_ih does not change the slope of the lift curve but it shifts it up or down. This effect can also be attained by incorporating it directly in R404/1545 which I recommend. It would not be consistent to treat this horizontal stabilizer offset separately while its lift slope (as a function of AoA) must be included in R404/1545 anyway.
In addition, a *htail_incidence* angle of zero degrees makes the effect disappear in MSFS while in the real world there could be an effect due to downwash.

However, R413 has a useful function: The total lift curve may shift up or down with Mach number in reality (change of zero lift AoA at high Mach number). This CL offset value can be set in R413 while at the same time *htail_incidence* then should be increased to 57.3 degrees to have it per radian. Be sure to set R420 to zero or the additional pitch moment will be huge.

$$CL_{\delta e} = \left(\frac{\partial R1539\_CL_{\delta e\_basic} + \partial R410\_CL_{\delta e\_M}}{\partial \delta e}\right)$$
$$* \delta e\ [elevator\_up\ OR\ down\_limit * R517\_Elev\_defl\_limit\_scalar\_vs\_\bar{q}]$$

This is the lift due to elevator deflection. Elevator deflection limits in aircraft.cfg are a function of R517. Also see chapter 0.

$$CL_{\delta s} = \frac{0.5 * \partial R1539\_CL_{\delta s}}{\partial \delta s} * \left(\delta s_{left} + \delta s_{right}\right)[spoiler\_limit]$$

This is the lift due to flight or roll spoilers deflection. (Negative) lift is calculated individually for left and right spoilers. R1539_CL_ds is the sum of both spoilers.

$$CL_{\alpha dot} = \left(\frac{\partial R1541\_CL_{\alpha dot\_basic} + \partial R411\_CL_{\alpha dot\_M}}{\partial \alpha dot}\right) * \frac{\alpha dot * MAC}{2 * V}$$

This is the lift due to AoA rate i.e. the change in AoA.

$$CL_q = \left(\frac{\partial R1541\_CL_{q\_basic} + \partial R412\_CL_{q\_M}}{\partial \frac{q * MAC}{2 * V}} \right) * \frac{q * MAC}{2 * V}$$

This is the lift due to pitch rate q.

5.3.3 Drag Coefficient

The total drag coefficient CD at a certain flight situation is calculated as follows. Note that in MSFS CD acts along the flight path (α) from side view, but along the X-body-axis when viewed from above.

$$CD_{Total} = CD0 + CDi + CD\beta + CD_{\delta f} + CD_{\delta s} + CD_{\delta r} + CD_{gear} + CD_r + CD_p$$

For simplicity (when calculating CG effects) CD0+CDi +CD_df is summarized in CD_wf (= wing, flaps)

$$CD_{wf} = CD0 + CDi + CD_{\delta f}$$

Where:

$$CD0 = (R1540\_CD0_{basic} + R430\_CD0_M) * parasite\_drag\_scalar$$

This is the minimum drag at clean configuration.

- CD0 decreases with side slip angle (Beta).
- In FSX Acceleration Pack instead of R430 the new non-linear table R154a_CD0_M can be used. It overrides R430.

$$CDi = \frac{CL\_linearized^2}{wing\ aspect\ ratio * oswald\_efficiency\_factor * pi} * induced\_drag\_scalar$$

This is the lift induced drag component. See Appendix 2 for explanation.

$$CD_{\delta f} = \frac{\partial R1540\_CD_{\delta f}}{\partial \delta f} * \sum_{n=0} (\delta f_n * drag\_scalar_n)$$

This is the drag component due to the individual deflection of each [flaps.n] section in aircraft.cfg and multiplied by the individual *drag_scalar* of each section.

$$CD_{\delta s} = \left(0.5 * \frac{\partial R1540\_CD_{\delta s}}{\partial \delta s}\right) * \left(\delta s_{left} + \delta s_{right}\right)[spoiler\_limit]$$

This drag is generated individually for left and right spoiler.

- If roll spoilers are extended only the left or right one causes drag. R1540_CD_ds is the sum of both spoilers.

$$CD_{gear} = \frac{\partial R1540\_CD_{gear}}{\partial gear\_pos} * gear\_pos$$

This drag component is generated due to the extended gear.

- Gear_pos: Retracted = 0, extended = 1

5.3.4 Side Force

The total side force coefficient CY, acting along the Y-body-axis, at a certain flight situation is calculated as follows:

$$CY_{Total} = CY_{\beta} + CY_{\delta r} + CY_{r} + CY_{p}$$

Where:

$$CY_{\beta} = \left(\frac{\partial R1542\_CY_{\beta\_basic} + \partial R440\_CY_{\beta\_M}}{\partial \beta}\right) * \beta$$

This is the side force due to sideslip angle.

$$CY_{\delta r} = \left(\frac{\partial R1542\_CY_{\delta r\_basic} + \partial R441\_CY_{\delta r\_M}}{\partial \delta r}\right) * \delta r\ [rudder\_limit * R519\_Rudder\_defl\_limit\_scalar\_vs\_\bar{q}]$$

This is the side force due to rudder deflection. Rudder deflection limit in aircraft.cfg is a function of R519.

$$CY_{r} = \left(\frac{\partial R1542\_CY_{r\_basic} + \partial R442\_CY_{r\_M}}{\partial \frac{r * wing\_span}{2 * V}}\right) * \frac{r * wing\_span}{2 * V}$$

This is the side force due to yaw rate r.

$$CY_{p} = \left(\frac{\partial R1542\_CY_{p\_basic} + \partial R443\_CY_{p\_M}}{\partial \frac{p * wing\_span}{2 * V}}\right) * \frac{p * wing\_span}{2 * V}$$

This is the side force due to roll rate p.

5.4 Propulsion Forces

Engines contribute to forces and moments according to their net thrust and their relative position to the current CG. Propeller engines also cause multiple more effects. While aerodynamic forces are a function of dynamic pressure, propulsion effects also act at zero flight speed. For this reason they are expressed here as absolute forces or moments instead of transforming them in a non-dimensional coefficient.

Thrust (T) is considered net thrust in lbs. which is directly provided by an MSFS parameter. In FSX and P3D it is possible to define an offset of the thrust vector:

- *Thrustanglespitchheading.n* can be added under [generalenginedata] in aircraft.cfg.

- First parameter is thrust vector pitch angle offset from body (positive=thrust vector downwards, in degree), second is heading offset (positive=thrust vector to the right, in degree). Bold letters indicate which of the two parameters is relevant in the context of the formula.
- Note the typo in the SDK: It is *Thrustangle**s**pitchheading*, not *Thrustanglepitchheading.*

Propulsion forces are summarized along the body axis (X, Y, Z).

$$F_{X\,Propulsion} = \sum_{n=0} \{T_n * [\cos(Thrustangles\boldsymbol{pitch}heading.n) + \cos(Thrustanglespitch\boldsymbol{heading}.n)]\}$$

$$F_{Y\,Propulsion} = \sum_{n=0} \{T_n * [\cos(Thrustangles\boldsymbol{pitch}heading.n) + \sin(Thrustanglespitch\boldsymbol{heading}.n)]\}$$

$$F_{Z\,Propulsion} = \sum_{n=0} \{T_n * [\sin(Thrustangles\boldsymbol{pitch}heading.n) + \sin(Thrustanglespitch\boldsymbol{heading}.n)]\}$$

6 Calculation of Moments

6.1 Summary

Pitch, roll and yaw moments act around the CG and along the three body axes X, Y, Z.
They consist of two components: The nominal Cm, Cl, Cn values from the AIR-file plus a correction for CG offsets which are based on the corresponding CL, CD and CY values. This means that any CG shift away from the reference location has an effect on the moments in MSFS.

Moments around lateral body axis (positive=nose pitching up):

$$M_{Total} = M_{Aero} + M_{Propulsion}$$

Moments around longitudinal body axis (positive=rolling to the right):

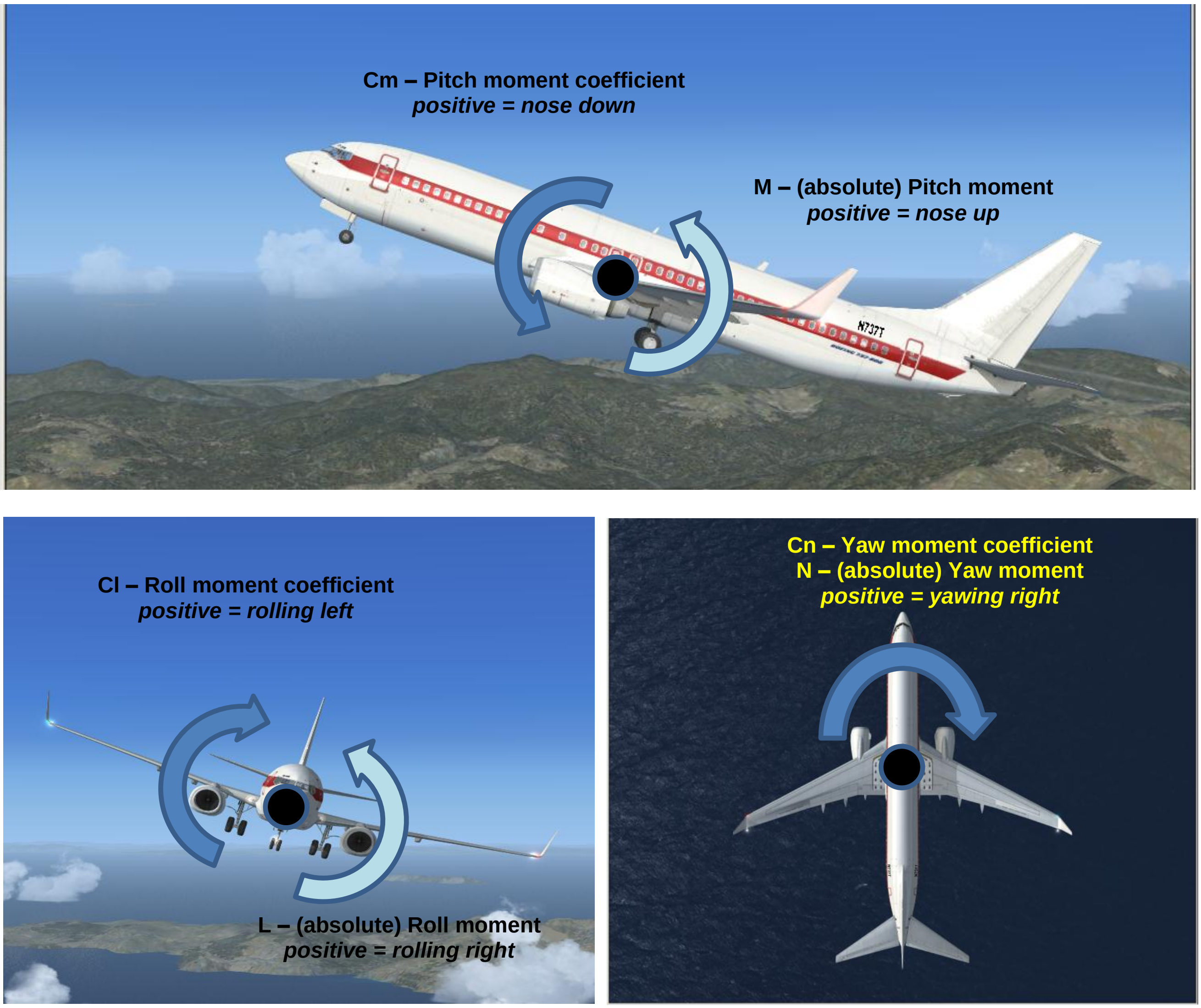
$$L_{Total} = L_{Aero} + L_{Propulsion}$$

Moments around vertical body axis (positive=yawing to the right):

$$N_{Total} = N_{Aero} + N_{Propulsion}$$

6.2 Axes System

The three moments act around the current CG along the X, Y, Z body axes. The sign convention for the pitch and roll moment coefficients differ from traditional theory. However, the absolute moments M, L, N used in this document are expressed in the traditional way.



6.3 Aerodynamic Moments

6.3.1 Overview

MSFS uses non-dimensional values from the AIR-file and scalars from aircraft.cfg which are multiplied by dynamic pressure ($\bar{q}$), wing_area and MAC or wing span to get absolute values for aerodynamic effects. There are pitch Moments (M, Cm), roll moments (L, Cl) and yaw moments (N, Cn) which act around the current CG along the X, Y, Z body axis.

$$M_{Aero} = -Cm_{Total} * \bar{q} * wing\_area * MAC$$

$$L_{Aero} = -Cl_{Total} * \bar{q} * wing\_area * wing\_span$$

$$N_{Aero} = Cn_{Total} * \bar{q} * wing\_area * wing\_span$$

As described in chapter 2 the nominal values in the AIR-file apply for the following nominal CG locations along the body axes:

Name	**Longitudinal (X)**	**Lateral (Y)**	**Vertical (Z)**
Pitch Moments	Wing Aerodynamic Center (AC) (set via Rec1534_3 or calculated by MSFS) $\Delta CG_{lon_{AC}}$	-	VMO $\Delta CG_{vert_{VMO}}$
Roll Moments	-	VMO $\Delta CG_{lat_{VMO}}$	VMO $\Delta CG_{vert_{VMO}}$
Yaw Moments	VMO $\Delta CG_{lon_{VMO}}$	VMO $\Delta CG_{lat_{VMO}}$	-

The symbols e.g. $\Delta CG_{vert\ VMO}$ are used in the formulas as offset value from the current CG to the reference location (VMO or AC) in feet. The calculation of the moment coefficient due to CG offset is shown in blue color.

The composition of the individual coefficients with reference to the parameters in the AIR-file and aircraft.cfg is described in detail in the following chapters.

6.3.2 Pitch Moments due to Aerodynamic Effects

The total pitch moment coefficient Cm at a certain flight situation is calculated as follows.

$$Cm_{Total} = Cm_{\alpha=0} + Cm_{\alpha} + Cm_{\delta f} + Cm_{ih} + Cm_{\delta e} + Cm_{\delta etr} + Cm_{\delta s} + Cm_{\alpha dot} + Cm_{q}$$

Where:

$$Cm_{\alpha=0} = (R1541\_Cm_{\alpha=0\ basic} + R433\_Cm_{\alpha=0\ M}) * \big((PitchMomentZeroAlpha\_Scalar - 1) * (1 - general\ realism\ scalar) + 1\big) + PitchMomentZeroAlpha\_Offset * (1 - general\ realism\ scalar)$$

This is the pitch moment at zero AoA. It adds to the Cm_α curve (R473/1546).

- R473/1546 may already contain a Cm offset at zero AoA. Then this parameter is actually Cm at zero lift.
- PitchMomentZeroAlpha is from aircraft.cfg under [realismconstants] and consists of a scalar and an offset (an additive value to Cm). Both depend on the general realism slider (inverse and linear).
  - General realism scalar = 0 (slider left): Full effect
  - General realism scalar = 1 (slider right): No effect, PitchMomentZeroAlpha_Scalar is 1 and Offset is 0

$$Cm_{\alpha} = \frac{\partial R473/1546\_Cm_{\alpha}}{\partial \alpha}[\alpha]$$

This is the basic pitch moment a function of the Cm vs. AoA curve in R473/1546.

- This curve is for the complete aircraft (wing-body-tail).
- A Cm offset at zero AoA (or zero Cm) can be incorporated to be consistent with R404/1545_CL_α where the CL offset is included too.

$$Cm_{\delta f} = \frac{\partial R1541\_Cm_{\delta f}}{\partial \delta f} * \sum_{n=0} (\delta f_n * pitch\_scalar_n)$$

This is the pitch moment due to the individual deflection of each [flaps.n] section in aircraft.cfg and multiplied by the individual *pitch_scalar* of each section.

For simplicity when calculation CG effect the above two terms can be summarized in Cm_α_wf (= wing, flaps):

$$\begin{aligned} Cm_{\alpha\, wf} &= Cm_{\alpha} + Cm_{\delta f} \\ &+ \frac{\Delta CG_{lon_{AC}}}{MAC} * \left(CL_{\alpha\, wf} * \cos(\alpha) + CD_{wf} * \sin(\alpha)\right) \\ &+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * \left(-CL_{\alpha\, wf} * \sin(\alpha) + CD_{wf} * \cos(\alpha)\right) \end{aligned}$$

$$\begin{aligned} Cm_{ih} &= \left(\frac{\partial R1541\_Cm_{ih\_basic} + \partial R423\_Cm_{ih\_M}}{\partial ih}\right) * htail\_incidence * R537\_htail\_pitch\_eff\_vs\_AoA \\ &+ \frac{\Delta CG_{lon_{AC}}}{MAC} * (CL_{ih} * \cos(\alpha)) \\ &+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * (-CL_{ih} * \sin(\alpha)) \end{aligned}$$

This is the pitch moment due to the incidence angle of the horizontal stabilizer. *htail_incidence* is a fixed value and does not change in-flight. It is not related to stab trim.

Recommendation:
R1541_Cm_ih does not change the slope of the pitch moment curve; it shifts it up or down. This effect can also be attained by incorporating it directly in R473/1546 which I recommend. It would not be consistent to treat this horizontal stabilizer offset separately while its pitch moment slope (as a function of AoA) must be included in R473/1546 anyway.
In addition, a *htail_incidence* angle of zero degrees makes the effect disappear in MSFS while in the real world there could be an effect due to downwash.

R423 Cm_ih_M should then be set to zero. Any Mach related offsets can be set in R433 instead.

$$
\begin{aligned}
Cm_{\delta e} = \Bigg\{ \Bigg[\left(\frac{\partial R1541\_Cm_{\delta e\_basic} + \partial R420\_Cm_{\delta e\_M}}{\partial \delta e} \right) \\
* \delta e \, [elevator\_up\ OR\ down\_limit * R517\_Elev\_defl\_limit\_scalar\_vs\_\bar{q}] \\
+ \frac{\delta e}{elevator\_up\_limit} * \frac{\partial R1541\_Cm_{\delta etr}}{\partial etr} * \delta etr \Bigg] * R536\_Elev\_pitch\_eff\_vs\_AoA \\
+ \frac{\Delta CG_{lon_{AC}}}{MAC} * (CL_{\delta e} * \cos(\alpha)) \\
+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * (-CL_{\delta e} * \sin(\alpha)) \Bigg\} \\
* R341\_Elev\_pitch\_eff\_vs\_deflection * elevator\_effectiveness
\end{aligned}
$$

This is the pitch moment due to elevator deflection.

- Elevator deflection limits in aircraft.cfg are a function of R517.
- Also see chapter 0 for an explanation of the third line in above equation.

$$
\begin{aligned}
Cm_{\delta etr} = \frac{\partial R1541\_Cm_{\delta etr}}{\partial etr} * \delta etr \, [elevator\_trim\_limit] * R536\_Elev\_pitch\_eff\_vs\_AoA \\
* R1525\_Elev\_trim\_pitch\_eff\_vs\_\bar{q} * elevator\_trim\_effectiveness
\end{aligned}
$$

This is the pitch moment due to elevator trim or horizontal stabilizer deflection.

- See chapter 0 for a detailed explanation.
- There is no lift or drag component for trim which means that a CG offset does not change Cm_detr.

$$
\begin{aligned}
Cm_{\delta s} = -\left(0.5 * \frac{\partial R1541\_Cm_{\delta s}}{\partial \delta s}\right) * \left(\delta s_{left} + \delta s_{right}\right)[spoiler\_limit] \\
+ \frac{\Delta CG_{lon_{AC}}}{MAC} * (CL_{\delta s} * \cos(\alpha) + CD_{\delta s} * \sin(\alpha)) \\
+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * (-CL_{\delta s} * \sin(\alpha) + CD_{\delta s} * \cos(\alpha))
\end{aligned}
$$

This is the pitch moment due to flight or roll spoilers deflection. Pitch moment is generated individually for left and right spoiler. R1541_Cm_ds is the sum of both spoilers.

$$
\begin{aligned}
Cm_{gear} = \frac{\partial R1541\_Cm_{gear}}{\partial gear\_pos} * gear\_pos \\
+ \frac{\Delta CG_{lon_{AC}}}{MAC} * \left(CD_{gear} * \sin(\alpha)\right) \\
+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * \left(CD_{gear} * \cos(\alpha)\right)
\end{aligned}
$$

This pitch moment component is generated due to the extended gear.

- Gear_pos: Retracted = 0, extended = 1

$$Cm_q = \left(\frac{\partial R1541\_Cm_{q\_basic} + \partial R422\_Cm_{q\_M}}{\partial \frac{q * MAC}{2 * V}}\right) * \frac{q * MAC}{2 * V} * R539\_Cm_q\_scalar\_vs\_AoA * pitch\_stability$$
$$+ \frac{\Delta CG_{lon_{AC}}}{MAC} * \left(CL_q * \cos(\alpha)\right)$$

This is the pitch moment due to pitch rate (pitch damping).

- The CG offset calculation above (red color) could not fully be verified.

$$Cm_{\alpha dot} = \left(\frac{\partial R1541\_Cm_{\alpha dot\_basic} + \partial R421\_Cm_{\alpha dot\_M}}{\partial \alpha dot}\right) * \frac{\alpha dot * MAC}{2 * V} * R538\_Cm_{\alpha dot}\_scalar\_vs\_AoA$$
$$+ \frac{\Delta CG_{lon_{AC}}}{MAC} * (CL_{\alpha dot} * \cos(\alpha))$$
$$+ \frac{\Delta CG_{vert_{VMO}}}{MAC} * (-CL_{\alpha dot} * \sin(\alpha))$$

This is the pitch moment due to AoA rate i.e. the change in AoA per second.

- The CG offset calculation above (red color) could not fully be verified.

6.3.3 Roll Moments due to Aerodynamic Effects

The total roll moment coefficient Cl at a certain flight situation is calculated as follows:

$$Cl_{Total} = Cl_\beta + Cl_\alpha + Cl_{\alpha\,wf} + CL_{ih} + Cl_{\delta e} + Cl_{\delta a} + Cl_{\delta s} + Cl_{\delta atr} + Cl_{\delta r} + Cl_{gear} + Cl_p + Cl_r$$

Where:

$$Cl_\beta = \beta * \left[\left(\frac{\partial R1543\_Cl_{\beta\_basic} + \partial R450\_Cl_{\beta\_M}}{\partial \beta}\right) * R451\_Cl_\beta\_scalar\_vs\_AoA[general\,realism\,scalar]\right.$$
$$* ((RollMomentFromBeta\_Scalar - 1) * general\,realism\,scalar + 1)$$
$$\left. + RollMomentFromBeta\_Offset * general\,realism\,scalar\right]$$
$$+ \frac{\Delta CG_{vert_{VMO}}}{wing\_span} * CY_\beta$$

This is the basic roll moment due to side slip.

- R451_Cl_β_scalar_vs_AoA is active only if the general realism slider is <u>fully</u> right, otherwise it is assumed to be 1.
- *RollMomentFromBeta* is from aircraft.cfg under [realismconstants] and consists of a scalar and an offset (an additive value to Cl_Beta per rad). Both depend on the general realism slider (linear).
    - General realism scalar = 0 (slider left): No effect, RollMomentFromBeta_Scalar is 1 and Offset is 0
    - General realism scalar = 1 (slider right): Full effect

Cl_{α}

For: Beta < -2°and Beta > 2°:

$$Cl_{\alpha} = R1538\_Cl\_vs\_AoA * hi\_alpha\_on\_roll * general\,realism\,scalar^2$$

For: -2°< Beta < 2°:

$$Cl_{\alpha} = R1538\_Cl\_vs\_AoA * hi\_alpha\_on\_roll * general\,realism\,scalar^2 * \frac{57.3}{2} * \beta[rad]$$

This is an additional roll moment coefficient as a function of AoA. It is mainly intended to enforce banking at stall AoA.

- The value in R1538 is effective if the side slip angle (Beta) is below -2° or above 2°.
- From Beta +/-2° to 0° the R1538 effect decreases linearly to zero.
- *Hi_alpha_on_roll* is from [flight_tuning] section.
- This value depends on the general realism slider setting in the MSFS menu. Its effect is a square function of the corresponding realism scalar value in fs9.cfg/fsx.cfg.
    - General realism scalar = 0 (slider left): No effect
    - General realism scalar = 0.5 (slider mid): 25% effect
    - General realism scalar = 1 (slider left): Full effect

$$Cl_{\alpha\,wf} = \frac{\Delta CG_{lat_{VMO}}}{wing\_span} * -CL_{\alpha\,wf} * \cos(\alpha)$$

This is the roll moment due to the wing and flaps lift. It only exists if CG has a lateral offset from VMO.

$$Cl_{ih} = \frac{\Delta CG_{lat_{VMO}}}{wing\_span} * -CL_{ih} * \cos(\alpha)$$

This is the roll moment due to the (fixed) incidence angle of the horizontal stabilizer (not related to stabilizer trim). It only exists if CG has a lateral offset from VMO.

$$Cl_{\delta e} = \frac{\Delta CG_{lat_{VMO}}}{wing\_span} * -CL_{\delta e} * \cos(\alpha)$$

This is the roll moment due to the elevator deflection. It only exists if CG has a lateral offset from VMO.

$$\begin{aligned} Cl_{\delta a} = {} & (\delta a_{left} + \delta a_{right})[aileron\_up\ OR\ down\_limit * R518\_Ail\_defl\_limit\_scalar\_vs\_\bar{q}] \\ & * \left(\frac{0.5 * \partial R1543\_Cl_{\delta a\_basic} + \partial R453\_Cl_{\delta a\_M}}{\partial \delta a}\right) * R342\_Ail\_roll\_eff\_vs\_defl * \\ & * R546\_Ail\_roll\_eff\_vs\_load\_factor * R1535\_Ail\_roll\_eff\_vs\_AoA * aileron\_effectiveness \\ & * [(RollMomentFromAilerons\_Scalar - 1) * (1 - general\,realism\,scalar) + 1] \\ & + RollMomentFromAilerons\_Offset * (1 - general\,realism\,scalar) \end{aligned}$$

This is the roll moment due to ailerons deflection.

- da_left and da_right (opposite direction) have both the same sign in MSFS.

- Aileron deflection limits in aircraft.cfg are a function of R518.
- Both ailerons reach their respective up/down deflection limit at the same full yoke position.
- The value in R453_Cl_da_vs_Mach is per single aileron (left aileron down) while R1543_CL_da is for both ailerons deflected (left down, right up).
- *RollMomentFromAilerons* is from aircraft.cfg under [realismconstants] and consists of a scalar and an offset (an additive value to Cl_da per rad). Both depend on the general realism slider (inverse and linear).
    - General realism scalar = 0 (slider left): Full effect
    - General realism scalar = 1 (slider right): No effect, *RollMomentFromAilerons*_Scalar is 1 and Offset is 0

$$Cl_{\delta s} = \frac{\partial R1543\_Cl_{\delta s}}{\partial \delta s} * \delta s\ [spoiler\_limit]$$
$$+ \frac{\Delta CG_{lat_{VMO}}}{wing\_span} * (CL_{\delta s} * \cos(\alpha) + CD_{\delta s} * \sin(\alpha))$$

Where

$$\delta s = (-\delta a_{left} - min\_ailerons\_for\_spoilerons) * aileron\_to\_spoileron\_gain$$

This is the roll moment due to roll spoiler deflection.

- Spoiler deflection is a linear function of aileron deflection.
- The value at R1543_Cl_ds is for the left spoiler which deflects upwards and causing left banking.

$$Cl_{\delta atr} = \frac{\partial R516\_Cl_{\delta atr}}{\partial atr} * \delta_{atr} * aileron\_trim\_effectiveness$$

This is the roll moment due to aileron trim.

- In MSFS it is an independent value not related to ailerons, meaning the ailerons do not move.
- The maximum trim angle is always 10 degrees.

$$Cl_{\delta r} = \left(\frac{\partial R1543\_Cl_{\delta r\_basic} + \partial R452\_Cl_{\delta r\_M}}{\partial \delta r}\right) * \delta r\ [rudder\_limit * R519\_Rud\_defl\_limit\_vs\_\bar{q}]$$
$$* \cancel{R343\_Rud\_eff\_vs\_rud\_defl}$$
$$+ \frac{\Delta CG_{vert_{VMO}}}{wing\_span} * -CY_{\delta r}$$

This is the roll moment due to rudder deflection. Note that R343 has never been working in any MSFS version.

$$Cl_{gear} = \frac{\Delta CG_{lat_{VMO}}}{wing\_span} * -CD_{gear} * \sin(\alpha)$$

This is the roll moment due to the extended gear. It only exists if the CG has a lateral offset from VMO and if AoA is <> 0.

$$Cl_p = \left(\frac{\partial R1543\_Cl_{p\_basic} + \partial R455\_Cl_{p\_M}}{\partial \frac{p * wing\_span}{2 * V}} \right) * \frac{p * wing\_span}{2 * V} * R456\_Cl_p\_scalar\_vs\_AoA[general\ realism\ scalar]$$
$$* roll\_stability$$
$$+ \frac{\Delta CG_{vert_{VMO}}}{wing\_span} * CY_p$$

This is the roll moment due to roll rate p (roll damping).

- R456_Cl_p_scalar_vs_AoA is active only if the general realism slider is fully right, otherwise it is assumed to be 1.

$$Cl_r = \left(\frac{\partial R1543\_Cl_{r\_basic} + \partial R454\_Cl_{r\_M}}{\partial \frac{r * wing\_span}{2 * V}} \right) * \frac{r * wing\_span}{2 * V}$$
$$+ \frac{\Delta CG_{vert_{VMO}}}{wing\_span} * CY_r$$

This is the roll moment due to yaw rate r.

6.3.4 Yaw Moments due to Aerodynamic Effects

The total yaw moment coefficient Cn at a certain flight situation is calculated as follows:

$$Cn_{Total} = Cn_\beta + Cn_\alpha + Cn_{\delta a} + Cn_{\delta r} + Cn_{\delta rtr} + Cn_p + Cn_r$$

Where:

$$Cn_\beta = \left(\frac{\partial R1544\_Cn_{\beta\_basic} + \partial R459\_Cn_{\beta\_M}}{\partial \beta} \right) * \beta * R460\_Cn_\beta\_scalar\_vs\_AoA[general\ realism\ scalar]$$
$$+ \frac{\Delta CG_{lon_{VMO}}}{wing\_span} * -CY_\beta$$

This is the basic yaw moment due to side slip.

- R460_Cn_Beta_scalar_vs_AoA is active only if the general realism slider is fully right; otherwise R460 is assumed to be 1.

Cn_α

For Beta < -2°and beta > 2°:

$$Cn_\alpha = R1537\_Cn\_vs\_AoA * hi\_alpha\_on\_yaw * general\ realism\ scalar^2$$

For -2°< beta < 2°:

$$Cn_{\alpha} = R1537\_Cn\_vs\_AoA * hi\_alpha\_on\_yaw * general\,realism\,scalar^2 \; * \frac{57.3}{2} * \beta[rad]$$

This is an additional yaw moment as a function of AoA. It is mainly intended to enforce yawing at stall AoA.

- The value in R1537 is effective if the side slip angle (Beta) is below -2° or above 2°.
- From Beta +/-2° to 0° the R1537 effect decreases linearly to zero.
- *Hi_alpha_on_yaw* is from [flight_tuning] section.
- This value depends on the general realism slider setting in the MSFS menu. Its effect is a square function of the corresponding realism scalar value in fs9.cfg/fsx.cfg.
    - General realism scalar = 0 (slider left): No effect
    - General realism scalar = 0.5 (slider mid): 25% effect
    - General realism scalar = 1 (slider left): Full effect

$$Cn_{\delta a} = \left(\frac{\partial R1544\_Cn_{\delta\alpha\_basic} + \partial R462\_Cn_{\delta\alpha\_M}}{\partial\delta\alpha}\right) * \left(\delta\alpha_{left} + \delta\alpha_{right}\right)[aileron\_up\ OR\ down\_limit * R518\_Ail\_defl\_limit\_vs\_\bar{q}] * \frac{1}{16} * general\,realism\,scalar$$

This is the yaw moment due to ailerons deflection.

- This value depends on the general realism slider setting in the MSFS menu. Its effect is a square function of the corresponding realism scalar value in fs9.cfg/fsx.cfg and scaled by 1/16.
    - General realism scalar = 0 (slider left): No effect
    - General realism scalar = 0.5 (slider mid): 50% effect
    - General realism scalar = 1 (slider left): Full effect (=1/16)
- This means that this SD should be set <u>16x higher</u> than the real value in the AIR-file to have the full nominal effect in MSFS.

$$Cn_{\delta r} = \left(\frac{\partial R1544\_Cn_{\delta r\_basic} + \partial R461\_Cn_{\delta r\_M}}{\partial\delta r}\right) * \cos(\beta) * \delta r\ [rudder\_limit * R519\_Rud\_defl\_limit\_vs\_\bar{q}] * \cancel{R343\_Rud\_eff\_vs\_rud\_defl} * rudder\_effectiveness + \frac{\Delta CG_{lon_{VMO}}}{wing\_span} * -CY_{\delta r} * \cos(\beta)$$

This is the yaw moment due to rudder deflection. Note that R343 has never been working in any MSFS version.

- The rudder effect decreases with β although CY_dr grows linearly with β.

$$Cn_{\delta rtr} = \frac{\partial R516\_Cn_{\delta rtr}}{\partial rtr} * \delta_{rtr} * rudder\_trim\_effectiveness$$

This is the yaw moment due to rudder trim.

- In MSFS it is an independent value not related to rudder, meaning the rudder does not move.
- The maximum trim angle is always 10 degrees.

$$Cn_p = \left(\frac{\partial R1544\_Cn_{p\_basic} + \partial R465\_Cn_{p\_M}}{\partial \frac{p * wing\_span}{2 * V}} \right) * \frac{p * wing\_span}{2 * V}$$
$$+ \frac{\Delta CG_{lon_{VMO}}}{wing\_span} * CY_p$$

This is the yaw moment due to roll rate.

$$Cn_r = \left(\frac{\partial R1544\_Cn_{r\_basic} + \partial R463\_Cn_{r\_M}}{\partial \frac{r * wing\_span}{2 * V}} \right) * \frac{r * wing\_span}{2 * V}$$
$$* R464\_Cn_r\_scalar\_vs\_AoA[general\ realism\ scalar]$$
$$* yaw\_stability$$
$$+ 2 * \frac{(vtail\_pos\_lon - CG_{lon_{VMO}})}{wing\_span} * Cn_{\delta r} * \delta r * \frac{r * wing\_span}{2 * V} * \sin(\beta) * ?$$
$$+ \frac{\Delta CG_{lon_{AC}}}{wing\_span} * -CY_r$$

This is the yaw moment due to yaw rate (yaw damping).

- R464_Cn_r_scalar_vs_AoA is active only if the general realism slider is <u>fully</u> right, otherwise it is assumed to be 1.
- The red line is supposed to be 'yaw damping due to rudder deflection'. The parameter *vtail_pos_lon* in the [airplane_geometry] section actually has a (minor) effect on the yaw moment as long a yaw rate exists and Cn_dr is not zero. Furthermore, the effect depends on the current side slip angle: At Beta=0° it is zero and then growths in a non-linear way until slowly disappearing again at Beta=90°. The above formula did not quite match the measured values, but gives about the correct dimension.

6.4 Propulsion Moments

6.4.1 Overview

Engines contribute to forces and moments according to their net thrust and their relative position to the current CG. Propeller engines also cause multiple more effects. While aerodynamic forces are a function of dynamic pressure, propulsion effects also act at zero flight speed. For this reason I will express them as absolute forces or moments instead of transforming them in a non-dimensional coefficient.

6.4.2 Pitch Moments due to Propulsion Effects

The total pitch moment (M) due to propulsion effects at a certain flight situation is calculated as follows:

$$M_{Propulsion} = M_{Thrust} + M_{Prop\ gyro} + M_{\delta e\_propwash} + M_{q\_propwash}$$

Where:

$$M_{Thrust} = \sum_{n=0}[T_n * \cos(Thrustanglespitchheading.n) * \Delta Eng.n_{vert_{CG}}$$
$$+ T_n * \sin(Thrustanglespitchheading.n) * \Delta Eng.n_{lon_{CG}}]$$

This is the pitch moment due to net thrust T of engine n.

- ΔEng.n $_{vert\ CG}$ is the vertical offset of the engine n in ft. from the current CG
- ΔEng.n $_{lon\ CG}$ is the longitudinal offset of the engine n in ft. from the current CG

$$M_{Prop\ gyro} = T_n * gyro\_precession\_on\_pitch * \beta dot * gyro\ realism\ scalar *?$$

This is the pitch moment due to gyro precession of propellers. *Above formula could not be completed.*
Gyro_precession_on_pitch from aircraft.cfg is a function of side slip change (βdot).

- Positive values = pitch up when Beta increases (reversed for counter clockwise propellers).
- This effect depends on the gyro realism slider setting in MSFS. It is a function of the corresponding realism scalar value in fs9.cfg/fsx.cfg.
  - Gyro realism scalar = 0 (slider left): No effect
  - Gyro realism scalar = 1 (slider right): Full effect
- This parameter also works in FS2004 although it is not mentioned in the FS2004 SDK.

$$M_{\delta e\_propwash} = \frac{\partial R1541\_Cm_{\delta e\_propwash}}{\partial \delta e} * \delta e\ [elevator\_up\ OR\ down\_limit * R517\_Elev\_defl\_limit\_vs\_\bar{q}]$$
$$* wing\_area * MAC * \bar{q}_{propwash} *?$$

This is the propwash effect on elevator. *Above formula could not be completed because there were deviations from the expected values*.

- $q_{propwash}$ is the additional dynamic pressure induced by the propellers = 0.5*rho*ΔV_n^2
- ΔV_n is the propeller induced additional airspeed (TAS) for the sum of engines, ft/s.

$$\Delta V_n = -V + \sqrt{V^2 + 2 * \frac{\sum T_n}{0.25 * pi * D^2 * \rho}}$$

- V is the airspeed of the plane (TAS), ft/s.
- T is the propeller net thrust for engine n, lbs.
- D is the propeller diameter, ft.
- The slower the plane's airspeed the higher the effect. It also exists at zero speed.
- MSFS does not account for engine to elevator distance, hstab span etc. All these influencing effects should be incorporated in Cm_de_propwash by the designer.
- Conditions for the effect:
  - Propeller planes only.
  - Engine lon position: Ahead of visual model origin.
  - Engine lat position: exactly at 0.0 ft. -> no effect for twin or 4-engines planes.

$$M_{q\_propwash} = \frac{\partial R1541\_Cm_{q\_propwash}}{\partial \frac{q * MAC}{2 * V}} * \frac{q * MAC}{2 * V} * \cos(\alpha)$$

This is supposed to be the pitch moment due to pitch rate caused by propwash (pitch damping). But flight tests did not correspond:

- This R1541 parameter seems not to depend on propwash or any other engine related effects. It even works with jet engines (running or shut down).
- It behaves like the regular Cm_q, but its effect decreases with cos (α).
- The realism sliders have no influence.

6.4.3 Roll Moments due to Propulsion Effects

The total roll moment (L) due to propulsion effects at a certain flight situation is calculated as follows:

$$L_{Propulsion} = L_{Thrust} + L_{Prop\ torque}$$

Where:

$$L_{Thrust} = \sum_{n=0} \Big[T_n * -\sin(Thrustangles\boldsymbol{pitch}heading.n) * \Delta Eng.n_{lat_{CG}} + T_n * -\sin(Thrustanglespitch\boldsymbol{heading}.n) * -cos(Thrustangles\boldsymbol{pitch}heading.n) * \Delta Eng.n_{vert_{CG}}\Big]$$

This is the roll moment due to net thrust T of engine n.

- ΔEng.n $_{\text{lat CG}}$ is the lateral offset of the engine n in ft. from the current CG.
- ΔEng.n $_{\text{vert CG}}$ is the vertical offset of the engine n in ft. from the current CG.

$$L_{Prop\ torque} = \sum shaft\ torque_n * \left(1 - \frac{v\ in\ kts\ (TAS)}{flaps\_up\_stall\_speed * 1.5}\right) * rotation * torque\_on\_roll * torque\ realism\ scalar$$

This is the roll moment due to propeller torque.

- It applies to propeller planes only.
- The roll effect is at its maximum at zero airspeed. It decreases linearly with airspeed (TAS) and becomes zero at *flaps_up_stall_speed* (TAS) from [reference Speeds] multiplied by 1.5. The effect remains zero above that speed.
- Shaft torque is per engine n, in lb-ft.
- The torque effect on roll moment depends on the rotation of the propeller(s) which can be individually set in aircraft.cfg set under [propeller]:
  - *Rotation = 1*: propeller no. 1 turns clockwise from pilot's view (→ left banking)
  - *Rotation = -1:* propeller no. 1 turns counter-clockwise from pilot's view (→ right banking)
  - *Rotation = 1,-1:* propeller no. 1 turns clockwise, propeller no. 2 counter-clockwise from pilot's view (→ i.e. no effect if both engines are running)
  - Note: Counter clockwise rotation can lead to problems with the visual effects of the propeller. For example, in FS2004 the default C172SP works fine, while in FSX the propeller of the default C172SP does not turn with the correct speed (gauges show right values).
- *torque_on_roll* in [flight_tuning] is a linear scalar on the roll moment due to propeller torque. Default=1. Must be positive to have any effect.
- This effect depends in a linear way on the torque realism slider setting in MSFS.
  - torque realism scalar = 0 (slider left): No effect

    - torque realism scalar = 1 (slider right): Full effect
- *Note: Gyro_precession_on_roll* is a parameter found in some FS2004 planes and listed in the FS2004 SDK but no effect could be seen. The string does not exist in the sim1.dll. In FSX the parameter was removed from the planes and the SDK.

6.4.4 Yaw Moments due to Propulsion Effects

The total yaw moment (N) due to propulsion effects at a certain flight situation is calculated as follows:

$$N_{Propulsion} = N_{Thrust} + N_{Prop\ p-factor} + N_{Prop\ gyro} + N_{\delta r\_propwash} + N_{r\_propwash}$$

Where:

$$N_{Thrust} = \sum_{n=0} [T_n * \sin(Thrustanglespitch\mathbf{heading}.n) * \cos(Thrustangles\mathbf{pitch}heading.n) * \Delta Eng.n_{lon_{CG}} - T_n * \cos(Thrustanglespitch\mathbf{heading}.n) * \cos(Thrustangles\mathbf{pitch}heading.n) * \Delta Eng.n_{lat_{CG}}]$$

This is the yaw moment due to net thrust T of engine n.

- ΔEng.n $_{lon\ CG}$ is the longitudinal offset of the engine n in ft. from the current CG
- ΔEng.n $_{lat\ CG}$ is thelateral offset of the engine n in ft. from the current CG

$$N_{Prop\ p-factor} = p\_factor\_on\_yaw * \alpha * ? * p\text{-}factor\ realism\ scalar$$

This is the yaw moment due to the p-factor. *Above formula is not complete.*

- *P_factor_on_yaw* is a function of AoA and only effective at positive AoA values.
- Positive values of p*_factor_on_yaw* lead to a left yawing at positive AoA values.
- *P_factor_on_yaw* depends on the p-factor realism slider setting in MSFS. Its effect is a function of the corresponding realism scalar value in fs9.cfg/fsx.cfg.
    - P-factor realism scalar = 0 (slider left): No effect
    - P-factor realism scalar = 1 (slider right): Full effect

$$N_{Prop\ gyro} = gyro\_precession\_on\_yaw * \alpha dot * ? * gyro\ realism\ scalar$$

This is the yaw moment due to propeller gyro precession. *Above formula is not complete.*

- *Gyro_precession_on_yaw* is a function of the rate of AoA change (αdot).
- It depends on the gyro realism slider setting in MSFS. Its effect is a function of the corresponding realism scalar value in fs9.cfg/fsx.cfg.
    - Gyro realism scalar = 0 (slider left): No effect
    - Gyro realism scalar = 1 (slider right): Full effect

$$N_{\delta r\_propwash} = \frac{\partial R1544\_Cn_{\delta r\_propwash}}{\partial \delta r} * \delta r\ [rudder\_limit * R519\_Rud\_defl\_limit\_vs\_\bar{q}] * wing\_area * wing\_span * \bar{q}_{propwash} * ?$$

This is the propwash effect on rudder. *Above formula could not be completed because there were deviations from the expected values*.

- $q_{propwash}$ is the additional dynamic pressure induced by the propellers = $0.5*rho*\Delta V_n^2$
  - ΔV_n is the propeller induced additional airspeed (TAS) for the sum of engines, ft/s

$$\Delta V_n = -V + \sqrt{V^2 + 2 * \frac{\sum T_n}{0.25 * pi * D^2 * \rho}}$$

- V is the airspeed of the plane (TAS), ft/s
- T is the propeller net thrust for engine n, lbs
- D is the propeller diameter, ft
- The slower the plane's airspeed the higher the effect. It also exists at zero speed.
- MSFS does not account for engine to elevator distance, vstab span etc. All these influencing effects should be incorporated in Cn_dr_propwash by the designer.
- Conditions for the effect:
  - Propeller planes only
  - Engine lon position: Ahead of visual model origin
  - Engine lat position: exactly at 0.0 ft. -> no effect for twin or 4-engines planes

$$N_{r\_propwash} = \frac{\partial R1544\_Cn_{r\_propwash}}{\partial \frac{r * wing\_span}{2 * V}} * \frac{r * wing\_span}{2 * V} * ?$$

This is the yaw moment due to yaw rate caused by propwash (yaw damping). *Above formula is not complete.* This is supposed to be the yaw moment due to yaw rate caused by propwash (yaw damping).

Appendix 1 Calculation of Wing Aerodynamic Center by MSFS

The following formula only applies if R1534 does not exist in an AIR-file. Otherwise the wing aerodynamic center longitudinal position can directly be set in R1534_3 (which is recommended).

MSFS will determine the nominal position for pitch moments in the AIR-file based on weight, pitch and lift parameters.

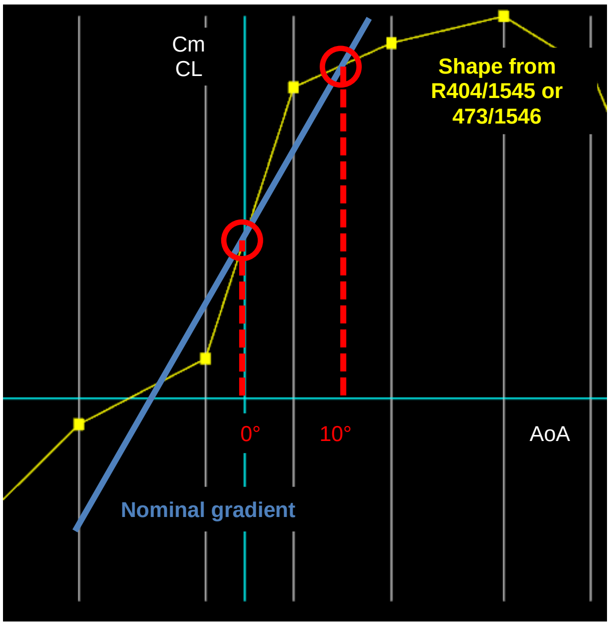
First, it computes the longitudinal CG position at zero fuel weight according to the aircraft.cfg parameters. Secondly, from this point a longitudinal offset is applied based on the relationship of the linearized $\frac{\partial Cm}{\partial \propto}$ (R473/1546) to the linearized $\frac{\partial CL}{\partial \propto}$ (R404/1545). This is the static stability in percent MAC. Multiplied by MAC it gives the distance in feet between the aerodynamic center of the plane (not wing) e.g. 45% MAC and the location for which the values in R473/1546 were designed (e.g. 25% MAC).

This means that in absence of R1534 the SD longitudinal reference point for pitch moments is always exactly in the middle of the zero fuel weight CG position and the plane's aerodynamic center. A typical setup could be ZFW at 5% MAC, SD reference point at 25% and the plane's aerodynamic center at 45%.

$$
\begin{aligned}
&Wing\ aerodynamic\ center\ (longitudinal\ offset\ from\ VMO, ft) \\
&= reference\_datum\_position_{lon} \\
&+ \frac{empty\_weight\_cg\_position_{lon} * empty\_weight + \sum(station\_load.n_{lon} * nominal\ station\_load.n\ weight)}{empty\_weight + \sum nominal\ station\_load.n\ weight} \\
&- \frac{Cm\_linearized}{CL\_linearized} * MAC
\end{aligned}
$$

where

reference_datum_position $_{lon}$	from aircraft.cfg in feet, longitudinal offset from visual model origin
empty_weight	from aircraft.cfg in lbs
empty_weight_cg_position $_{lon}$	from aircraft.cfg, longitudinal position in feet
station_load.n lon $_{pos}$	from aircraft.cfg in lbs, longitudinal position of station_load.0, station_load.1 etc.
nominal station_load.n weight	from aircraft.cfg in lbs, nominal weight means the weight defined in aircraft.cfg, not the actual one set in MSFS menu via load editor
Cm_linearized	from R473/1546. Since pitch moment vs. AoA is a linear function only in a small AoA range, MSFS looks up Cm in R473/1546 at AoA=0° and AoA=10° and calculates the gradient from these two points (see figure below)
CL_linearized	from R404/1545, same principle as described above for R473/1546. Values is from simulation variable in the FSX SDK: LINEAR CL ALPHA



Example

Input

empty_weight_cg_position = 5, 0, -1
empty_weight = 100000 lbs
station_load.0 = 11000, 15, 0, 0, Payload1
station_load.1 = 15000, -12, 0, 0, Payload2
R473/1545 = Cm gradient between 0° and 10° AoA = 1.5
R404/1545 = CL gradient between 0° and 10° AoA = 5
MAC = 12 feet
reference_datum_position = 20, 0, 4

Output

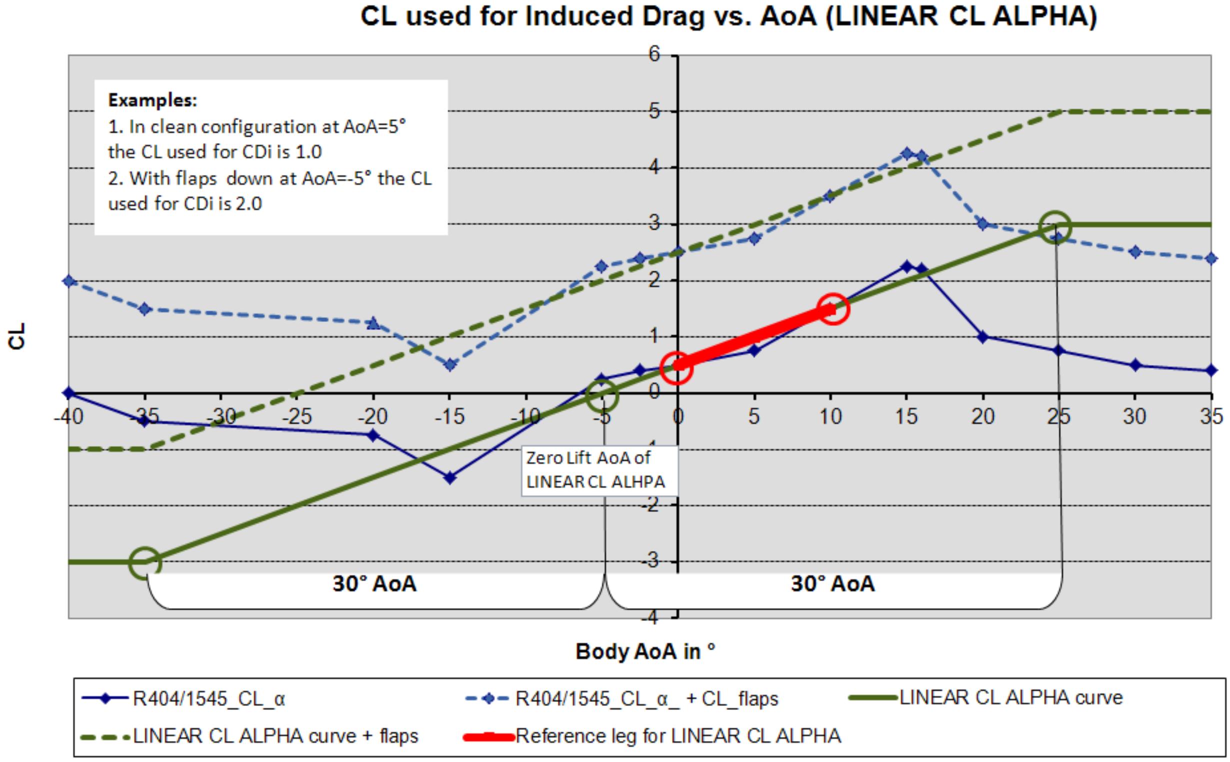
$$Wing\ aerodynamic\ center\ (longitudinal\ offset\ from\ VMO, ft)$$
$$= \frac{(5 * 100000\ +\ 15 * 11000 - 12 * 15000)}{(100000 +\ 11000 + 15000)} - \frac{1.5}{5} * 12 - 20 =\ -16.4\ feet$$

In this example adding R1534 and setting the lon parameter to -16.4 feet would give the same result in MSFS.

Appendix 2 Calculation of Induced Drag

The following explanation is my theory of CDi computation in MSFS and it is – like anything else here – open for discussion.

Induced drag coefficient (CDi) is calculated by MSFS almost the same way as found in most text books, but with some simplification. Instead of using the true lift coefficient (CL) at a certain AoA, MSFS looks up the CL value from an assumed linear CL_α slope, called LINEAR CL ALPHA, as described now.



1. Lift curve construction

a) LINEAR CL ALPHA is calculated by MSFS as follows: From the CL vs. AoA curve in R404/1545 MSFS draws a linear line between CL at AoA=0° and CL at AoA=10°. This part of the slope defines LINEAR CL ALPHA. It does not matter if there are any explicitly defined points at 0° or 10° AoA, MSFS just interpolates to get the CL values at 0 and 10°. All other data points are ignored for CDi computation.
→LINEAR CL ALPHA is a simulation variable found in the FSX SDK and can also be displayed in gauges (*A: LINEAR CL ALPHA, degrees* OR *radians*). Here radians are used.

$$LINEAR\ CL\ ALPHA = \frac{R404/1545\_CL_{\alpha=10°} - R404/1545\_CL_{\alpha=0}}{\frac{10°}{57.3}}$$

b) The crossing point of the LINEAR CL ALPHA curve and CL=0 is the AoA used as zero lift reference, called ZERO LIFT ALPHA. From there the LINEAR CL ALPHA line is extended further to the left by a delta of -30° AoA and to the right by +30°. Outside this 60° range the LINEAR CL ALPHA curve remains constant which means that CDi also remains constant.
→ZERO LIFT ALPHA is also a simulation variable found in the FSX SDK. It is valid for zero flaps.

c) When flaps are extended, the LINEAR CL ALPHA curve moves parallel upwards. This only applies to flaps, not to other lift suppliers such as elevator or spoilers.

d) The CL Mach factor (R401) is not taken into account for CDi calculation. LINEAR CL ALPHA does not change its slope with Mach number. This gives an incorrect (normally too low) CL because the actual CL_α curve is steeper at high Mach no. This can partially be corrected by R430 CD0 vs. Mach and in combination with Oswald efficiency factor.

2. In-flight CDi calculation

In-flight MSFS takes the current AoA and then looks up the corresponding CL value from the LINEAR CL ALPHA curve. It also adds CL_flaps if deflected. The resulting CL_linearized is taken for CDi calculation.

$$CDi = \frac{CL\_linearized^2}{wing\ aspect\ ratio * oswald\_efficiency\_factor * pi} * induced\_drag\_scalar$$

Where *CL_linearized* is:

For the normal AoA range:
-30°< (α – ZERO LIFT ALPHA) < 30°

$$CL\_linearized = LINEAR\ CL\ ALPHA * \frac{\alpha - ZERO\ LIFT\ ALPHA}{57.3} + CL_{\delta f}$$

For high AoA:
(α – ZERO LIFT ALPHA) > 30°

$$CL\_linearized = LINEAR\ CL\ ALPHA * \frac{30°}{57.3} + CL_{\delta f}$$

For high negative AoA:
(α – ZERO LIFT ALPHA) < -30°

$$CL\_linearized = LINEAR\ CL\ ALPHA * \frac{-30°}{57.3} + CL_{\delta f}$$

3. Conclusion of CDi calculation

- MSFS CDi may differ from traditional calculation if the true CL_α curve is not linear or if AoA is high.
- Mach effect on total drag must be considered separately for designing. This requires some tweaking of CD0, R430 and Oswald efficiency factor (or even CD_flaps). It is possible to achieve an exact fit with real world data for at least a typical cruise flight level/speed, some clean low altitude/speed situations and for flaps configurations. But there will be small deviations in total drag at intermediate levels that cannot be corrected.

Appendix 3 Aircraft.cfg Parameters

Based on SDK FSX SP2.

Mandatory: In case a mandatory parameter does not exist, MSFS adds it when the aircraft is loaded for the first time.
Fallback source: If a mandatory parameter is omitted in the aircraft.cfg, MSFS looks up the equivalent value in the AIR-file. If neither exists, a standard value is used (normally 0 or 1)

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
[WEIGHT_AND_BALANCE]					
max_gross_weight	Lbs	FS9+FSX	Yes	Calculated	
empty_weight	Lbs	FS9+FSX	Yes	R1101_8 – station_load.n	
reference_datum_position	Feet	FS9+FSX	Yes	-	
empty_weight_CG_position	Feet	FS9+FSX	Yes	-	
max_number_of_stations	-	FS9+FSX	No	-	Max allowed number of stations is very high, more than 300 stations (limit not reached in testing).
station_load.*n*	-	FS9+FSX	No	-	Numbering must start at 0 and must be continuous.
station_name.*n*	-	FSX	No	-	Name can also be included at the end of station_load.n. Numbering must start at 0 and must be continuous.
empty_weight_roll_MOI	$Slugs*ft^2$	FS9+FSX	Yes	R1001_1	The values should be provided for zero fuel weight, not empty weight. See chapter 3.2.
empty_weight_pitch_MOI	$Slugs*ft^2$	FS9+FSX	Yes	R1001_2	Same as above.
empty_weight_yaw_MOI	$Slugs*ft^2$	FS9+FSX	Yes	R1001_3	Same as above.
empty_weight_coupled_MOI	$Slugs*ft^2$	FS9+FSX	Yes	R1001_4	Same as above.
CG_forward_limit	%MAC/ 100	FS9+FSX	No	-	No effect on flight model observed, only used for gauges.
CG_aft_limit	%MAC/ 100	FS9+FSX	No	-	No effect on flight model observed, only used for gauges.
[fuel]					

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
fuel_type	Flag	FS9+FSX	Yes	-	
Center1 *and all other tanks*	-	FS9+FSX	No	-	
number_of_tank_selectors	-	FS9+FSX	No	-	
electric_pump	BOOL	FS9+FSX	No	-	
fuel_dump_rate	% per second	FSX	No	-	
manual_transfer_pump	BOOL	FS9+FSX	No	-	
engine_driven_pump	BOOL	FSX	No	-	
anemometer_pump	BOOL	FS9+FSX	No	-	
[airplane_geometry]					
wing_area	Sq. feet	FS9+FSX	Yes	R1204_1	Nominal value from airplane manufacturer.
wing_span	Feet	FS9+FSX	Yes	R1204_2	Nominal value from airplane manufacturer.
wing_root_chord	Feet	FS9+FSX	Yes	R1515_4	See chapter 3.1.2. for determination of the correct value.
wing_cg_ref_chord	Feet	FS9+FSX	No	R1515_1	MAC length for gauge display. Does only affect readout variable for CG position, but has no physical effect on flight dynamics.
wing_dihedral	Degree	FS9+FSX	Yes	R1204_4	No effect observed in FS2004/FSX. Used for FSEdit only.
wing_incidence	Degree	FS9+FSX	Yes	R1204_5	No effect observed in FS2004/FSX.
wing_twist	Degree	FS9+FSX	Yes	R1204_6	No effect observed in FS2004/FSX.
wing_winglets_flag	Boolean	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
oswald_efficiency_factor	-	FS9+FSX	Yes	Calculated from wing aspect ratio	Oswald efficiency factor for the complete aircraft. Should be considered together with CD0, R430 Wave Drag and R401 Lift vs. Mach. A value different from real world data may be chosen to get a better fit with MSFS drag calculation (see Appendix 2).
wing_sweep	Degrees	FS9+FSX	Yes	R1515_3	Used for setting MAC position.
wing_pos_apex_lon	Feet	FS9+FSX	Yes	R1515_2 & RDP_lon	Used for setting MAC position.
wing_pos_apex_vert	Feet	FS9+FSX	Yes	RDP_vert	Used for setting MAC position.
htail_area	Sq. feet	FS9+FSX	Yes	R1205_1	No effect observed in FS2004/FSX.
htail_span	Feet	FS9+FSX	Yes	R1205_2	No effect observed in FS2004/FSX.

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
htail_pos_lon	Feet	FS9+FSX	Yes	R1205_3& RDP_lon	No effect observed in FS2004/FSX.
htail_pos_vert	Feet	FS9+FSX	Yes	RDP_vert	No effect observed in FS2004/FSX.
htail_incidence	Degree	FS9+FSX	Yes	R1205_5	Works in conjunction with R1539_CL_ih, R1541_Cm_ih, R413, R423, R537. No effect if value is 0.
htail_sweep	Degree	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
vtail_area	Sq. feet	FS9+FSX	Yes	R1206_1	No effect observed in FS2004/FSX.
vtail_span	Feet	FS9+FSX	Yes	?	No effect observed in FS2004/FSX.
vtail_sweep	Degree	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
vtail_pos_lon	Feet	FS9+FSX	Yes	R1206_3 & RDP_lon	Affects yaw damping due to rudder deflection, but effect is minor (see chapter 6.3.4).
vtail_pos_vert	Feet	FS9+FSX	Yes	R1206_2 & RDP_vert	No effect observed in FS2004/FSX.
elevator_area	Sq. feet	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
aileron_area	Sq. feet	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
rudder_area	Sq. feet	FS9+FSX	Yes	-	No effect observed in FS2004/FSX.
elevator_up_limit	Degree	FS9+FSX	Yes	R320_4	Positive value, maximum elevator deflection, multiplied by R517 Elev. Deflection vs. dynamic pressure.
elevator_down_limit	Degree	FS9+FSX	Yes	R320_5	Positive value, maximum elevator deflection, multiplied by R517 Elev. Deflection vs. dynamic pressure.
aileron_up_limit	Degree	FS9+FSX	Yes	R320_6	Positive value, maximum aileron deflection, multiplied by R518 Aileron Deflection vs. dynamic pressure.
aileron_down_limit	Degree	FS9+FSX	Yes	R320_7	Positive value, maximum aileron deflection, multiplied by R518 Aileron Deflection vs. dynamic pressure.
rudder_limit	Degree	FS9+FSX	Yes	R320_8	Positive value, maximum rudder deflection, multiplied by R519 Rudder Deflection vs. dynamic pressure.
elevator_trim_limit	Degree	FS9+FSX	Yes	R320_9	Positive value, maximum pitch trim deflection. Up and down limits are equal, different limits can be set via FSUIPC offsets.
spoiler_limit	Degree	FS9+FSX	Yes	-	Positive value, maximum flight and roll spoiler deflection.
spoilerons_available	Boolean	FS9+FSX	Yes	R340_1	Roll spoilers (spoilerons) available. Flight spoilers (speed brakes) not affected.

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
aileron_to_spoileron_gain	-	FS9+FSX	Yes	R340_4	Roll spoilers deflection gain with aileron up deflection (e.g. value 2 = 1° of additional aileron up deflection deploys additional 2° of roll spoiler deflection).
min_ailerons_for_spoilerons	Degree	FS9+FSX	Yes	R340_3	Roll spoilers start to extend if aileron up deflection is at least this value.
min_flaps_for_spoilerons	Degree	FS9+FSX	Yes	R340_2	Roll spoilers become effective if the nominal flaps value is at least this value.
spoiler_handle_available	BOOL	FS9+FSX	No	-	
auto_spoiler_available	BOOL	FSX	Yes	R337	In FS2004 this is in the AIR-file in Rec 337.
flap_to_aileron_scale	-	FSX	No	-	
aileron_to_rudder_scale	-	FS9+FSX	No	-	
positive_g_limit_flaps_up	g units	FSX	Yes	R1101_13	In FS2004 this setting is in R1101 (1 g = 936 units).
positive_g_limit_flaps_down	g units	FSX	Yes	R1101_15	In FS2004 this setting is in R1101 (1 g = 936 units).
negative_g_limit_flaps_up	g units	FSX	Yes	R1101_14	In FS2004 this setting is in R1101 (1 g = 936 units).
negative_g_limit_flaps_down	g units	FSX	Yes	R1101_16	In FS2004 this setting is in R1101 (1 g = 936 units).
load_safety_factor	-	FSX	No	-	
fly_by_wire	BOOL	FSX	No	-	
[flight_tuning]					This section is for users who do not want to edit the AIR-File. All effects (and much more) can also be attained by adjusting the SD in the AIR-File. In this case all multipliers should be set to 1.
cruise_lift_scalar	-	FS9+FSX	No	-	Multiplier on the lift curve of the wing-body-tail (R404/1545) plus flaps.
parasite_drag_scalar	-	FS9+FSX	No	-	Multiplier on CD_0.
induced_drag_scalar	-	FS9+FSX	No	-	Multiplier on CDi.
elevator_effectiveness	-	FS9+FSX	No	-	Multiplier on Cm_de, but not on CL_de.
aileron_effectiveness	-	FS9+FSX	No	-	Multiplier on Cl_da.
rudder_effectiveness	-	FS9+FSX	No	-	Multiplier on Cl_dr, but not on CY_dr.
pitch_stability	-	FS9+FSX	No	-	Multiplier on Cm_q.
roll_stability	-	FS9+FSX	No	-	Multiplier on Cl_p.
yaw_stability	-	FS9+FSX	No	-	Multiplier on Cn_r.
elevator_trim_effectiveness	-	FS9+FSX	No	-	Multiplier on Cm_detr.

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
aileron_trim_effectiveness	-	FS9+FSX	No	-	Multiplier on Cl_datr.
rudder_trim_effectiveness	-	FS9+FSX	No	-	Multiplier on Cl_drtr.
hi_alpha_on_roll	-	FS9+FSX	No	-	This affects Cl_α.
hi_alpha_on_yaw	-	FS9+FSX	No	-	This affects Cn_α.
p_factor_on_yaw	-	FS9+FSX	No	-	This affects the yaw moment due to propeller p-factor.
torque_on_roll	-	FS9+FSX	No	-	Multiplier on the prop torque (roll) moment.
gyro_precession_on_yaw	-	FS9+FSX	No	-	This affects the yaw moment due to gyro precession.
gyro_precession_on_pitch	-	FS9+FSX	No	-	This affects the pitch moment due to gyro precession.
[flaps.*n*]					See chapter 0 for flaps details.
type	-	FS9+FSX	No	-	Type=1: Trailing edge Type=2: Leading edge
span-outboard	-	FS9+FSX	No	-	No effect on flight model observed.
extending-time	s	FS9+FSX	No	-	Time for full deflection/retraction of this flaps set.
flaps-position.*n*	-	FS9+FSX	No	-	Deflection angle in degree. Optionally a maximum speed limit for flaps travel can be set for each flaps position (kts). Above this speed the flaps cannot be moved. e.g. [flaps.0] Flaps-position.0=0 Flaps-position.1=5, 250 Flaps-position.2=10, 230
damaging-speed	kts	FS9+FSX	No	-	
damaging-speed-variation	kts?	FS9+FSX	No	-	Not mentioned in the SDK, from sim1.dll. Not tested.
blowout-speed	kts	FS9+FSX	No	-	
lift_scalar	-	FS9+FSX	No	-	Multiplier on CL_df
drag_scalar	-	FS9+FSX	No	-	Multiplier on CD_df
pitch_scalar	-	FS9+FSX	No	-	Multiplier on Cm_df
system_type	-	FS9+FSX	No	-	

Parameter	Unit	Sim Version	Manda-tory	Fallback	Description
[realismconstants]					
rollmomentfrombeta	-	FS9+FSX	No	-	This affects Cl_β.
rollmomentfromailerons	-	FS9+FSX	No	-	This affects Cl_da.
pitchmomentzeroalpha	-	FS9+FSX	No	-	This affects Cm at zero lift.
[reference speeds]					
flaps_up_stall_speed	Kts	FS9+FSX	Yes	R1101_2	Used for propeller torque: Max speed where torque has an effect on roll. Affects also AI traffic.
full_flaps_stall_speed	Kts	FS9+FSX	Yes	-	Affects AI traffic only. MSFS sets this to 0.8 x flaps_up_stall_speed.
cruise_speed	Kts	FS9+FSX	Yes	?	Affects AI traffic only.
max_indicated_speed	Kts	FS9+FSX	Yes	R1101_3	Triggers overspeed warning.
max_mach	Mach no.	FS9+FSX	Yes	R316	Triggers overspeed warning.

Appendix 4 AIR-File Parameters

Names conventions

Names of the main records (in capital letters) are from ASM2AIR.exe from the LM P3D SDK.

OBSOLETE_AIR
When an aircraft is (re)loaded MSFS always checks if all required parameters in the aircraft.cfg exist. If this is not the case it will first try to look up the values from the AIR-file (OBSOLETE_AIR_*) and if found will add the equivalent lines and values to the aircraft.cfg. If the obsolete AIR-file records do not exist, MSFS will add a default value (normally 0 resp. 1 for scalars).
This applies to mandatory lines in aircraft.cfg only. Optional parameters are not transferred from the AIR-file.

- Example: If *wing_area* does not exist when the aircraft is loaded, MSFS will look for R1204_1 and will add the line and the value to aircraft.cfg. If R1204 is not present, *wing_area* is set to 0.

UNUSED_AIR
These parameters are not read anymore. They were used in pre FS2004 and FSX versions and later replaced by entries in the aircraft.cfg.

Simulation version code:
61S = FS98
70 = FS2000
70S = FS2000 with Updates
80 = FS2002
10X = FSX Acceleration Pack

Offset	MSFS Name	Description
0001h	UNUSED_AIR_TITLE	
0002h	OBSOLETE_AIR_DESCRIPTION	
0003h	OBSOLETE_AIR_PERFORMANCE	
0004h	UNUSED_AIR_TAIL_NUMBER	
0100h	UNUSED_AIR_SIM_NAME	
0101h	UNUSED_AIR_NORMAL_MDL_NAME	
0102h	UNUSED_AIR_SIMPLE_MDL_NAME	

Offset	MSFS Name	Description
0103h	UNUSED_AIR_CRASH_MDL_NAME	
0104h	UNUSED_AIR_PANEL_NAME	
0105h	OBSOLETE_AIR_AC_CATEGORY	
0300h	OBSOLETE_AIR_CONTROL_RESPONSE	
0301h	OBSOLETE_AIR_CG_TO_EYEPOINT	
0302h	OBSOLETE_AIR_FUEL_CAPS	
0310h	OBSOLETE_AIR_ENGINE_TYPE	
0311h	OBSOLETE_AIR_NUMBER_ENGINES	
0312h	OBSOLETE_AIR_FUEL_WEIGHT	
0313h	OBSOLETE_AIR_THROTTLE_LIMIT	
0314h	OBSOLETE_AIR_LANDME_AVAIL	
0315h	OBSOLETE_AIR_FLAPS_DETENTS	
0316h	OBSOLETE_AIR_MAX_OPERATING_MACH	
0316h	OBSOLETE_MAX_OPERATING_MACH	
0317h	OBSOLETE_AIR_AUTOPILOT_AVAIL	
0318h	OBSOLETE_AIR_FLAPS_AVAIL	
0319h	OBSOLETE_AIR_STALL_HORN_AVAIL	
0320h	OBSOLETE_AIR_CONTROLS_CONSTANTS	
0321h	OBSOLETE_AIR_PROP_ADVANCE_AVAIL	
0322h	OBSOLETE_AIR_ENG_MIXTURE_AVAIL	
0323h	OBSOLETE_AIR_CARB_HEAT_AVAIL	
0324h	OBSOLETE_AIR_PITOT_HEAT_AVAIL	
0325h	OBSOLETE_AIR_GEAR_WARNING_HORN	
0327h	OBSOLETE_AIR_SPOILER_AVAIL	
0328h	OBSOLETE_AIR_IS_TAIL_DRAGGER	
0329h	OBSOLETE_AIR_AUTO_THROTTLE	
0330h	OBSOLETE_AIR_PROP_TYPE_AVAIL	
0331h	UNUSED_AIR_STROBES_AVAIL	

Offset	MSFS Name	Description
0332h	OBSOLETE_AIR_TOE_BRAKES_AVAIL	
0333h	UNUSED_AIR_NAV_OBS_AVAIL	
0334h	UNUSED_AIR_VOR2_GAUGE_AVAIL	
0335h	UNUSED_AIR_NAV_RADIOS_AVAIL	
0336h	UNUSED_AIR_GYRO_DRIFT_AVAIL	
0337h	OBSOLETE_AIR_AUTO_SPOILER_AVAIL	
0338h	OBSOLETE_AIR_AUTO_BRAKES_AVAIL	
0339h	OBSOLETE_AIR_STICK_SHAKER	
0340h	OBSOLETE_AIR_SPOILERONS	
0341h	AIR_ELEVATOR_SCALING	
0342h	AIR_AILERON_SCALING	
0343h	AIR_RUDDER_SCALING	No effect observed in MSFS (all versions).
0344h	UNUSED_AIR_SMOKE_AVAIL	
0400h	AIR_GROUND_EFFECT	
0401h	AIR_CL_MACH	
0402h	AIR_CL_BASIC	Probably never used. Maybe a linear value for CL_a.
0403h	AIR_CL_HIGH_AOA	Probably never used. Maybe a correction of CL_a as a function of AoA (stall).
0404h	AIR_CL_ALPHA	Lift vs. AoA (probably replacement for R402 and R403).
0410h	AIR_CL_DELTAE	
0411h	AIR_CL_ADOT	
0412h	AIR_CL_Q	
0413h	AIR_CL_IH	
0420h	AIR_CM_DELTAE	
0421h	AIR_CM_ADOT	
0422h	AIR_CM_Q	
0423h	AIR_CM_IH	
0430h	AIR_CD_O	
0431h	AIR_CL2_K	Probably never used.

Offset	MSFS Name	Description
0432h	AIR_CDI	Probably never used.
0433h	AIR_CMO	
0440h	AIR_CY_BETA	
0441h	AIR_CY_DELTAR	
0442h	AIR_CY_R	
0443h	AIR_CY_P	
0450h	AIR_CL_BETA	
0451h	AIR_ALPHA_ON_CL_BETA	
0452h	AIR_CL_DELTAR	
0453h	AIR_CL_DELTAA	
0454h	AIR_CL_R	
0455h	AIR_CL_P	
0456h	AIR_ALPHA_ON_CLP	
0459h	AIR_CN_BETA	
0460h	AIR_ALPHA_ON_CN_BETA	
0461h	AIR_CN_DELTAR	
0462h	AIR_CN_DELTAA	Set this value 16x higher because it is scaled by the general realism slider (at full left position).
0463h	AIR_CN_R	
0464h	AIR_ALPHA_ON_CNR	
0465h	AIR_CN_P	
0472h	AIR_THRUST_EFFICIENCY	Probably never used.
0473h	AIR_CM_ALPHA	
0500h	OBSOLETE_AIR_PROP_ENGINE	
0501h	UNUSED_AIR_PROP_MIX_PWR	
0502h	UNUSED_AIR_PROP_PWR_CORR	
0503h	UNUSED_AIR_PROP_TORQUE_2D	
0504h	UNUSED_AIR_PROP_PROP_2D	
0505h	OBSOLETE_AIR_61S_RECIPROCATING_ENGINE	

Offset	MSFS Name	Description
0506h	AIR_61S_VOLUMETRIC_EFFICIENCY	
0507h	AIR_61S_COMBUSTION_EFFICIENCY	
0508h	AIR_61S_ENG_MECHANICAL_EFFICIENCY	
0509h	AIR_61S_ENGINE_FRICTION	
0510h	OBSOLETE_AIR_61S_PROPELLER	
0511h	AIR_61S_PROP_EFFICIENCY	
0512h	AIR_61S_PROP_PWR_CF	
0513h	OBSOLETE_AIR_61S_HYDRAULICS	
0514h	UNUSED_AIR_61S_HYDRAULIC_PRESSURE_TABLE	
0515h	UNUSED_AIR_61S_AIL_RUD_TRIM_SCALES	
0516h	AIR_61S_AIL_RUD_TRIM_CONSTANTS	
0517h	AIR_61S_ELEVATOR_ELASTICITY	
0518h	AIR_61S_AILERON_ELASTICITY	
0519h	AIR_61S_RUDDER_ELASTICITY	
0520h	UNUSED_AIR_61S_ROCKET_ENGINE_CONSTANTS	
0521h	OBSOLETE_AIR_61S_GYROSCOPIC_CONSTANTS	
0522h	OBSOLETE_AIR_61S_P_FACTOR_CONSTANTS	
0523h	OBSOLETE_AIR_61S_GEAR_SYSTEM_TYPE	
0524h	OBSOLETE_AIR_61S_FLAP_SYSTEM_TYPE	
0525h	OBSOLETE_AIR_61S_FUEL_TANK_LEFT_MAIN	
0526h	OBSOLETE_AIR_61S_FUEL_TANK_RIGHT_MAIN	
0527h	OBSOLETE_AIR_61S_FUEL_TANK_LEFT_AUX	
0528h	OBSOLETE_AIR_61S_FUEL_TANK_RIGHT_AUX	
0529h	OBSOLETE_AIR_61S_FUEL_TANK_LEFT_TIP	
0530h	OBSOLETE_AIR_61S_FUEL_TANK_RIGHT_TIP	
0531h	OBSOLETE_AIR_61S_FUEL_TANK_CENTER_1	
0532h	OBSOLETE_AIR_61S_FUEL_TANK_CENTER_2	
0533h	OBSOLETE_AIR_61S_FUEL_TANK_CENTER_3	

Offset	MSFS Name	Description
0534h	OBSOLETE_AIR_61S_FUEL_TANK_EXTERNAL_1	
0535h	OBSOLETE_AIR_61S_FUEL_TANK_EXTERNAL_2	
0536h	AIR_61S_ALPHA_ON_CMDE	
0537h	AIR_61S_ALPHA_ON_CMIH	
0538h	AIR_61S_ALPHA_ON_CMADOT	
0539h	AIR_61S_ALPHA_ON_CMQ	
0540h	AIR_61S_EGT	
0541h	AIR_61S_CHT	
0542h	AIR_61S_RADIATOR_TEMPERATURE	
0543h	AIR_61S_OIL_TEMPERATURE	
0544h	AIR_61S_OIL_PRESSURE	
0545h	AIR_61S_FUEL_PRESSURE	
0546h	AIR_61S_AILERON_LOAD_FACTOR_EFF	
0600h	OBSOLETE_AIR_JET_ENGINE	
0601h	UNUSED_AIR_JET_FF_N1	
0602h	UNUSED_AIR_JET_N1_AF	
0603h	UNUSED_AIR_JET_N1_EPR	
0604h	UNUSED_AIR_JET_N1_N2	
0605h	UNUSED_AIR_JET_N2_FPR	
0606h	UNUSED_AIR_JET_EPR_EGT	
1000h	UNUSED_AIR_GLOBAL_VARS	
1001h	OBSOLETE_AIR_MOMENTS_OF_INERTIA	
1002h	OBSOLETE_AIR_ENGINE_POSITIONS	
1003h	OBSOLETE_AIR_FUEL_TANK_POSITIONS	
1004h	OBSOLETE_AIR_GEAR_INFORMATION	
1005h	OBSOLETE_AIR_SCRAPE_POSITIONS	
1100h	UNUSED_AIR_SIM_VARS	
1101h	AIR_VARIABLES	

Offset	MSFS Name	Description
1199h	AIR_AP_PID_CONTROLLERS	Used for AI planes only. Was used for autopilot PID up to FS2002.
1200h	UNUSED_AIR_AP_PID_CONTROLLERS	
1201h	UNUSED_AIR_AP_MAX_BANK_PITCH	
1202h	UNUSED_AIR_AP_ELEVTRM_IN_BANK	
1203h	UNUSED_AIR_AP_LIMITS	
1204h	OBSOLETE_AIR_WING_SPECS	
1205h	OBSOLETE_AIR_HSTAB_SPECS	
1206h	OBSOLETE_AIR_VSTAB_SPECS	
1300h	UNUSED_AIR_SOUND_CONTROL_FILE	
1400h	AIR_HELI_VERTICAL_TAIL	
1401h	AIR_HELI_HORIZONTAL_TAIL	
1402h	AIR_HELI_MAIN_ROTOR	
1403h	AIR_HELI_TAIL_ROTOR	
1404h	AIR_HELI_MISCELLANOUS	
1500h	UNUSED_AIR_SPITFIRE_CRC	
1501h	OBSOLETE_AIR_70_TURBINE_ENGINE	
1502h	AIR_70_N2_TO_N1_TABLE	
1503h	AIR_70_MACH_0_CORRECTED_COMMANDED_N2	
1504h	AIR_70_MACH_HI_CORRECTED_COMMANDED_N2	
1505h	AIR_70_CORRECTED_N2_FROM_FF	
1506h	AIR_70_N1_AND_MACH_ON_THRUST	
1507h	AIR_70_CORRECTED_AIRFLOW	
1508h	AIR_70_N1_TO_SHAFT_TORQUE	
1509h	OBSOLETE_AIR_70_PROP_FEATHERING	
1510h	OBSOLETE_AIR_70_PROP_SYNC	
1511h	OBSOLETE_AIR_70_PROP_REVERSE	
1512h	OBSOLETE_AIR_70_VACUUM_SYSTEM	
1513h	OBSOLETE_AIR_70_PNEUMATIC_SYSTEM	

Offset	MSFS Name	Description
1514h	OBSOLETE_AIR_70_TURBOPROP	
1515h	OBSOLETE_AIR_70_WING_SPECS_2	Important note: R1515_1 (first parameter) is directly read by MSFS if *wing_cg_ref_chord* in aircraft.cfg does not exist. The latter is an optional parameter and only exists if manually added to aircraft.cfg (see chapter 2.3). R1515 should be completely removed to avoid wrong CG position indications in gauges (however R1515_1 has no physical effect)
1516h	OBSOLETE_AIR_70_MACH_LIMITING	
1517h	OBSOLETE_AIR_70_CG_LIMITING	
1518h	OBSOLETE_AIR_70_PROP_DEICE	
1519h	OBSOLETE_AIR_70_STRUCTURAL_DEICE	
1520h	OBSOLETE_AIR_70_FUEL_PUMPS	
1521h	AIR_70_PRIMARY_NOZZLE	
1522h	AIR_70_REVERSER_NOZZLE	
1523h	AIR_70_VARIABLE_INLET	
1524h	AIR_70_AFTERBURNER_ON_THRUST_TABLE	
1525h	AIR_70_ELEVATOR_TRIM_ELASTICITY	
1526h	AIR_70_ITT	
1527h	UNUSED_AIR_70_PROP_CCW_ROTATE	
1528h	UNUSED_AIR_70_FDE_ENGINE_TUNING_SCALARS	
1529h	UNUSED_AIR_70_FDE_AERO_TUNING	
1530h	OBSOLETE_AIR_70_TORQUE_ON_FLIGHT_SCALAR	
1531h	UNUSED_AIR_70_FLAPS_NONLINEAR_TABLE	
1532h	AIR_70_EPR	
1533h	OBSOLETE_AIR_70_FLIGHT_DIRECTOR	
1534h	AIR_70_AERODYNAMIC_CENTER	Wing aerodynamic center offset from VMO. Only parameter 3 works. See chapter 2.4
1535h	AIR_70S_ALPHA_ON_CLDA	
1536h	AIR_70S_ALPHA_ON_CNDR	
1537h	AIR_70S_CN_ALPHA_YAW	
1538h	AIR_70S_Cl_ALPHA_ROLL	

Offset	MSFS Name	Description
1539h	AIR_80_LIFT_PARAMS	Overwrites R1101 values.
1540h	AIR_80_DRAG_PARAMS	Overwrites R1101 values if R1539 exists.
1541h	AIR_80_PITCH_PARAMS	Overwrites R1101 values if R1539 exists.
1542h	AIR_80_SIDE_FORCE_PARAMS	Overwrites R1101 values if R1539 exists.
1543h	AIR_80_ROLL_PARAMS	Overwrites R1101 values if R1539 exists.
1544h	AIR_80_YAW_PARAMS	Overwrites R1101 values if R1539 exists.
1545h	AIR_80_CL_ALPHA_TABLE	Overwrites R404.
1546h	AIR_80_CM_ALPHA_TABLE	Overwrites R473.
1547h	OBSOLETE_AIR_80_AERODYNAMIC_CENTER	Not used in FS2004 and FSX. Use R1534.
1548h	AIR_80_DENSITY_ON_TP_TORQUE	
1549h	AIR_10XPACK_N1_MACH_ON_NOZZLE	
154ah	AIR_10XPACK_CD0_MACH	

Disclaimer

This document is released as freeware. Under no circumstances may this file be sold for profit.
The document may not be uploaded to any websites or public servers without prior permission of the author.

The information in this document is intended to modify aircraft configuration files (mainly aircraft.cfg and .AIR-files) for both freeware and payware projects. The goal is to share knowledge about aircraft configuration tuning to improve flight behavior of add-on aircrafts.

This is not a Microsoft or Lockheed Martin document. Microsoft and Lockheed Martin were not involved in the creation or publication in any way. I am neither affiliated with those companies nor have I been in contact with them. This document is the result of a private project. All information is based on conducted test flights in Microsoft Flight Simulator and can be reproduced by any experienced user. No reverse engineering, decompiling or disassembling of copyrighted software took place.

Credits

The following **excellent** tools were used for exploring flight dynamics in MSFS:

- AFSD 4.0 (and pevoius versions) by Hervé Sors
- AirEd by William Roth
- Aired.ini by Ron Freimuth, Jerry Beckwith
- Aircraft Airfile Manager (AAM) by Karl-Heinz Klotz
- AAM.ini by Sergio di Fusco
- FSUIPC by Peter Dowson
- FS-Interrogate by Pelle F. S. Liljendal

Source of information:

- Microsoft Flight Simulator 2000/2002/2004/FSX SDK
- Microsoft ESP SDK at http://msdn.microsoft.com/en-us/library/cc526961.aspx
- Lockheed Martin Prepar3D 1.3 SDK at http://www.prepar3d.com/support/sdk/
- Various online documents covering flight dynamics such as:
  - Aircraft Simulation Techniques Used in Low-Cost, Commercial Software, Michael K. Zyskowski, Microsoft Corporation, Redmond, WA, 2003
    http://download.microsoft.com/download/1/7/8/17808bed-9b1e-4542-90a6-4c1d8af27fae/Aircraft_Sim_Tech_Zyskowski.pdf
    Basic explanation of how MSFS simulates the flight model.
  - The Simulation of a Jumbo Jet Transport Aircraft, D6-30643, C. Rodney Hanke and Donald R. Nordwall, 1970
    http://naca.larc.nasa.gov/search.jsp?R=19730001300&qs=Ns%3DLoaded-Date|0%26N%3D4294781323
    This includes a complete data set of all SD of a 747 in graphical form and allows a very precise modeling in MSFS!
  - Stability and Control Derivative Estimates Obtained from Flight Data for the Beech 99 Aircraft, Russel R. Tanner and Terry D. Montgomery
    http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19790011963_1979011963.pdf