AeroAircraft6DOFSS

Flight dynamics model of super-sonic aircraft

In[206]:=

```
<< C:\\Hopsan\Compgen\CompgenNG.mx
```

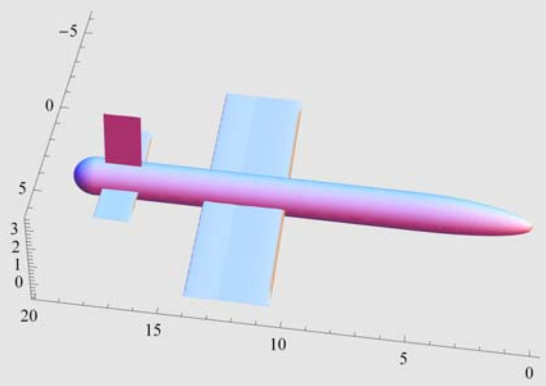
In[207]:=

```
Off[General::"spell1"]
```

In[208]:=

```
path = ToFileName[{"C:", "HopsanTrunk",
    "ComponentLibraries", "defaultLibrary", "Special", "AeroComponents"}];
```

Flight dynamics simulation is used in a wide range of applications from aircraft design to operations evaluation and flight training. This means that the models are to be used over very different time scales. For aircraft design very detailed flight dynamics characteristics needs to be evaluated, while much simplified models can be used for simualation of missions and to represent other planes in pilot training (and computer games). In order to meat these ends of the spectra the simulation models must incorporate a lot of detail while on the other hand be robust so that they can be used also with large time steps. Here the symbolic math packages *Mathematica* is used. By using symbolic manipulation high level differential algebraic equations can be transformed into low level code (C++) where highly robust numerical solvers are integrated into the model, for highly efficient robust simulation.



In[209]:=

```
domain = "Aero";
displayName = "Aircraft6DOFSS";
brief = "Flight dynamics model of super-sonic aircraft";
componentType = "ComponentC";
author = "Petter Krus <petter.krus@liu.se>";
affiliation = "Division of Fluid and Mechatronic Systems, Linköping University";
SetFilenames[path, domain, displayName];
ResetComponentVariables[];
Date[]
```

Out[217]=

```
{2015, 9, 18, 12, 43, 22.4153390}
```

Defining node variables

In[225]:=
```
MechanicCnode[n_, position_, comment_] := {n,
  {{f[n], 0., Real, "N", "force"},
   {x[n], position, Real, "m", "position"},
   {sx[n], 0., Real, "m/s", "speed"},
   {cx[n], 0., Real, "N", "wave variable"},
   {Zx[n], 0., Real, "N/m/s", "Char. impedancs"}}, "mechanicCnodes", comment}
```

Loading Library

In[226]:=
```
Derivative[1,0][Atan2][y_,x_]:=
D1Atan2[y,x];
```

In[227]:=
```
Derivative[1,0][Atan2L][y_,x_]:=
D1Atan2L[y,x];
```

In[228]:=
```
Derivative[0,1][Atan2][y_,x_]:=
D2Atan2[y,x];
```

In[229]:=
```
Derivative[0,1][Atan2L][y_,x_]:=
D2Atan2L[y,x];
```

In[230]:=
```
Derivative[1][ArcSinL][x_]:=
DArcSinL[x];
```

In[231]:=
```
Derivative[1][SecL][x_]:=
DxSecL[x];

Unprotect[Sec];
Sec[x_]:=SecL[x]
```

Component description

This model simulates a 3D model of an airplane.

Parameters and variables used in this component.

Declarations

Alias for some variables

In[234]:=
```
θ = Thetao;
ψ = Psi;
ϕ = Phi;
α = Alpha;
β = Beta;
Λ = Lambda1;
Γ = Gamma1;
```

In[241]:=
```
M_θ̇ = Mdvtheta;
M_ψ̇ = Mdvpsi;
M_ϕ̇ =.;
```

In[244]:=
```
q = qpress;
q0 =.;
q1 =.;
q2 =.;
q3 =.;
```

In[249]:=
```
```

Declarations of variables

The global output variables are those variables that are of interst outside the component.

In[250]:=

```
outputVariables = {
   {xcg, 0, double, "m", "Horizontal position 1"},
   {ycg, 0, double, "m", "Horizontal position 2"},
   {zcg, 0, double, "m", "Vertical position"},
   {vx, 0, double, "m", "Horizontal speed 1"},
   {vy, 0, double, "m", "Horizontal speed 2"},
   {vz, 0, double, "m", "Vertical speed"},
   {ψ, 0, double, "rad", "Azimuth angle"},
   {θ, 0, double, "rad", "Elevation angle"},
   {ϕ, 0, double, "rad", "Bank angle"},
   {Ub, 100, double, "m/s", "Speed xb-axis"},
   {Vb, 0, double, "m/s", "Speed yb-axis"},
   {Wb, 0, double, "m/s", "Speed zb-axis"},
   {Pb, 0, double, "rad/s", "Angular velocity"},
   {Qb, 0, double, "rad/s", "Angular velocity"},
   {Rb, 0, double, "rad/s", "Angular velocity"},
   {q0, 0, double, "", "quartenion 0"},
   {q1, 0, double, "", "quartenion 1"},
   {q2, 0, double, "", "quartenion 2"},
   {q3, 0, double, "", "quartenion 3"},
   {AlphaAttack, 0, double, "rad", "Angle of atack"},
   {BetaSlip, 0, double, "rad/s", "Sideslip angle"},
   {altitude, 0, double, "m", "altitude"},
   {gfx, 0, double, "m/s^2", "g-force in x"},
   {gfy, 0, double, "m/s^2", "g-force in y"},
   {gfz, 0, double, "m/s^2", "g-force in z"},
   {CL1, 0, double, "", "Lift coeff. wing 1"},
   {Cd1, 0, double, "", "Drag coeff. wing 1"},
   {Fax, 0, double, "N", "Aero force in z"},
   {Faz, 0, double, "N", "Aero force in x"}
  };
```

The input variables are those variables that are inputs during the simulation to the component. In practice thery are very similar to input parameters as the can be converted from each other in the Hopsan interface. It only effects the default settings in the component XML file.

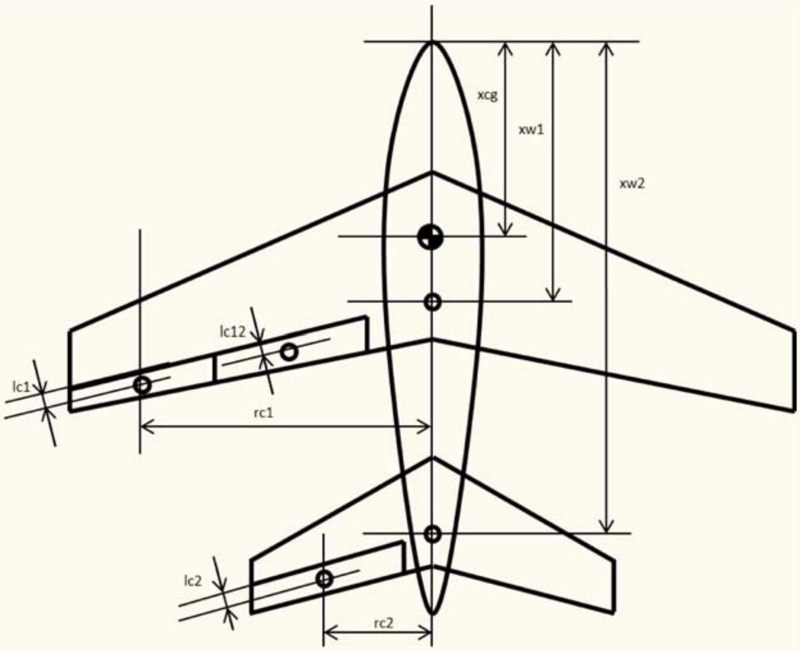
In[251]:=

```
inputVariables = {
   {thrustl, 0., double, "N", "Engine thrust"},
   {thrustr, 0., double, "N", "Engine thrust"},
   {dezthrustl, 0., double, "rad", "Thrust angle"},
   {dezthrustr, 0., double, "rad", "Thrust angle"},
   {deythrustl, 0., double, "rad", "Thrust angle"},
   {deythrustr, 0., double, "rad", "Thrust angle"},
   {Mfuel, 0., double, "kg", "Fuel weight"},
   {Mcargo, 0., double, "kg", "Cargo weight"},
   {rho, 1.25, double, "kg/m3", "Air density"},
   {vM, 340., double, "m/s", "Speed of sound"},
   {vturbx, 0., double, "m/s", "air turbulence x"},
   {vturby, 0., double, "m/s", "air turbulence y"},
   {vturbz, 0., double, "m/s", "air turbulence z"},
   {wturbx, 0., double, "rad/s", "air turbulence x"},
   {wturby, 0., double, "rad/s", "air turbulence y"},
   {wturbz, 0., double, "rad/s", "air turbulence z"}};
```

In[252]:=
```
nodeConnections = {
    MechanicRotCnode[al1, 0., 0., "mechanical node left airleron 1"],
    MechanicRotCnode[ar1, 0., 0., "mechanical node right airleron 1"],
    MechanicRotCnode[al12, 0., 0., "mechanical node left flaperon 1"],
    MechanicRotCnode[ar12, 0., 0., "mechanical node right flaperon 1"],
    MechanicRotCnode[al2, 0., 0., "mechanical node left airleraon 2"],
    MechanicRotCnode[ar2, 0., 0., "mechanical node right airleraon 2"],
    MechanicRotCnode[fin, 0., 0., "mechanical node fin"]};
```

inptutParameters are parameters that are normally used throughout the whole system.

The local parameters are the component specific parameters. Default parameters are based on the F-16.



Geometric data is entered in dimensionless form that relates to the actual data through the reference area and the aspect ratio.

In[253]:=
```
inputParameters = {
    {afin, 0.3, double, "rad", "break angle 1"},
    {an1, 0.6, double, "rad", "Neg. break angle 1"},
    {an2, 0.6, double, "rad", "Neg. break angle 2"},
    {ap1, 0.9, double, "rad", "Pos. break angle 1"},
    {ap2, 0.7, double, "rad", "Pos. break angle 2"},
    {AR1, 3.62, double, "", "Aspect ratio 1"},
    {AR2, 3.62, double, "", "Aspect ratio 2"},
    {ARfin, 1.5, double, "", "Aspect ratio fin"},
    {Cd01, 0.0045, double, "", "Drag coef. 1"},
    {Cd02, 0.0045, double, "", "Drag coef. 2"},
    {Cd0b, 0.03, double, "", "Drag coef. body"},
    {Cd0fin, 0.0045, double, "", "Drag coef. fin"},
    {CdW0, 0.048, double, "", "Wave drag coef."},
    {CLalpha1, 2.1, double, "", "L. slope coef. 1"},
    {CLalpha2, 2.2, double, "", "L. slope coef. 2"},
    {CLalphabh, 2., double, "", "L. slope c. body h"},
    {CLalphabv, 2., double, "", "L. slope c. bodyv"},
    {CLalphafin, 0.80, double, "", "L. sl. c. fin"},
    {CLde1, 0.1, double, "", "Ctrl surface coef 1"},
```

```
{CLde12, 0.2, double, "", "Flap rudder coef 1"},
{Cdide1, 0., double, "", "Flap rudder drag coef 1"},
{Cdide12, 0., double, "", "Flap rudder drag coef 1"},
{Cdide112, 0., double, "", "Flap rudder cross drag coef 1"},
{de10, 0.01, double, "", "rudder min drag angle 1"},
{de120, 0.01, double, "", "Flap min drag angle 1"},
{dM, 0.1, double, "", "Width of transonic region (rel. Mach)"},
{Cm01, -0.1, double, "", "Mom coeff. wing 1"},
{Cmfs1, -0.5, double, "", "Mom coeff.1, fully separated"},
{Cmde1, 0.02, double, "", "Mom slop coeff 1"},
{Cmde12, 0.1, double, "", "Flap Mom slop coeff 1"},
{CLdefin, 0.0827084`, double, "", "Rudder coef 1"},
{Cydeelev, 0.1, double, "", "elevator side force coef"},
{dah1, 1.0, double, "", "down wash effect on 1"},
{dah2, 0.6, double, "", "down wash effect on 2"},
{en, 2.71828, double, "", "e"},
{e1, 0.95, double, "", "Osw. effic. factor 1"},
{e2, 0.95, double, "", "Osw. effic. factor 21"},
{efin, 0.95, double, "", "Osw. eff. f. fin"},
{epsM, 0.3, double, "", "Mach smoothening factor"},
{awfin, .2, double, "", "CL exponent fin"},
{awn1, .2, double, "", "CL exponent neg. 1"},
{awn2, .2, double, "", "CL exponent neg. 2"},
{awp1, .2, double, "", "CL exponent pos 1"},
{awp2, .2, double, "", "CL exponent neg 1"},
{gamma1, -0.0872665`, double, "rad", "dehidral"},
{gamma2, -0.0872665`, double, "rad", "dehidral"},
{hthrust0, 0., double, "", "engine vert. pos"},
{ia1, 0., double, "rad", "incidence angle 1"},
{ia2, 0.02, double, rad "", "incidence angle 1"},
{Ix0, 0.0147, double, " ", "Norm. Inertia moment Ix/(Me S1)"},
{Ixz0, 0.0055, double, " ", "Norm. Inertia moment"},
{Iy0, 1.131, double, " ", "Norm. Inertia moment"},
{Iz0, 1.279, double, " ", "Inertia moment"},
{lambda1, 0.436332, double, "rad", "sweep 1"},
{lambda2, 0.436332, double, "rad", "sweep 2"},
{lambdafin, 0.785398, double, "rad", "sweep fin"},
{lc10, 0.01, double, "", "norm. ctrl surf. 1 ac fr hinge lc1/sqrt(AR1 S1)"},
{lc20, 0.05, double, "", "norm. ctrl surf. 2 ac fr hinge lc1/sqrt(AR1 S1)"},
{lc120, 0.01, double, "", "norm. flap 1 ac fr hinge"},
{lcfin0, 0.01, double, "", "ctrl s. fin ac fr hinge"},
{Me, 8700., double, "kg", "Empty weight"},
{mac0, 1., double, "", "mean aerodynamic cord/Sqrt(S1/b1)"},
{rc10, 0.25, double, "", "norm. ctrl surface 1 mom. arm"},
{rc120, 0.25, double, "", "norm. ctrl surface 12 mom. arm"},
{rc20, 0.15, double, "", "norm. ctrl surface 1 mom. arm"},
{rcfin0, .1, double, "", "norm. ctrl surf. fin mom. arm"},
{S1, 27., double, "m2", "wing area 1"},
{S20, 0.36, double, "", "norm. wing area 2"},
{Sbh0, 0.2, double, "", "norm. hor. proj. area"},
{Sbv0, 0.1, double, "", "norm.body vert. proj. area"},
{Sfin0, 0.17, double, "", "norm. fin area"},
{xbach0, 3, double, "", "norm. body ac. hor."},
{xbacv0, 3, double, " ", "norm. body ac vert."},
{xbcge0, 3, double, " ", "norm. body cg"},
{xcargo0, 3, double, " ", "norm. cargo pos."},
```

```
    {xfuel0, 3, double, " ", ""},
    {xw10, 3, double, " ", "norm. wing1  position"},
    {xw20, 4.8, double, " ", "norm. wing 2 position"},
    {xwfin0, 4.8, double, "", "norm. fin position"},
    {xeng0, 4.8, double, "", "norm. fin position"},
    {yeng0, 0., double, "", "engines off. from center"},
    {g0, 9.81, double, "m/s^2", "Gravity acceleration"},
    {kground, 10000., double, "N/m", "Ground stiffness (for limitiation)"},
    {cground, 1000., double, "Ns/m", "Ground damping (for limitiation)"}};
```

The Qnodes are the connections to other components

In[254]:=

```
Qnodes = {};
```

There are also a constants. Noiter is the number of iterations performed in each time step

Nomenclature

L_{BV} :Transformation matrix, body coordinates to vehicle-carried vertical frame

Component equations

Definitions

In[255]:=

$$\xi := \begin{pmatrix} 1 & \mathrm{Sin}[\phi]\,\mathrm{Tan}[\theta] & \mathrm{Cos}[\phi]\,\mathrm{Tan}[\theta] \\ 0 & \mathrm{Cos}[\phi] & -\mathrm{Sin}[\phi] \\ 0 & \mathrm{Sin}[\phi]\,\mathrm{SecL}[\theta] & \mathrm{Cos}[\phi]\,\mathrm{SecL}[\theta] \end{pmatrix}$$

The transformation matrices for transformation from body to earth axis are (Etkin (4.5,2)

In[256]:=

$$L_x[\phi\_] := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{Cos}[\phi] & \mathrm{Sin}[\phi] \\ 0 & -\mathrm{Sin}[\phi] & \mathrm{Cos}[\phi] \end{pmatrix}$$

In[257]:=

$$L_y[\theta\_] := \begin{pmatrix} \mathrm{Cos}[\theta] & 0 & -\mathrm{Sin}[\theta] \\ 0 & 1 & 0 \\ \mathrm{Sin}[\theta] & 0 & \mathrm{Cos}[\theta] \end{pmatrix}$$

In[258]:=

$$L_z[\psi\_] := \begin{pmatrix} \mathrm{Cos}[\psi] & \mathrm{Sin}[\psi] & 0 \\ -\mathrm{Sin}[\psi] & \mathrm{Cos}[\psi] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Transformation from vehicle-carried vertical frame to body coordinates can be expressed as (Etkin 4.5,3)

In[259]:=

$$L_{BV} := L_x[\phi].L_y[\theta].L_z[\psi]$$

In[260]:=

$$\mathrm{MatrixForm}[L_{BV}]$$

Out[260]//MatrixForm=

```
( Cos[Psi] Cos[Thetao]                                   Cos[Thetao] Sin[Psi]
  -Cos[Phi] Sin[Psi] + Cos[Psi] Sin[Phi] Sin[Thetao]     Cos[Phi] Cos[Psi] + Sin[Phi] Sin[Psi]
  Sin[Phi] Sin[Psi] + Cos[Phi] Cos[Psi] Sin[Thetao]      -Cos[Psi] Sin[Phi] + Cos[Phi] Sin[Psi
```

$$\begin{pmatrix} \mathrm{Cos}[\mathrm{Psi}]\,\mathrm{Cos}[\mathrm{Thetao}] & \mathrm{Cos}[\mathrm{Thetao}]\,\mathrm{Sin}[\mathrm{Psi}] & -\mathrm{Sin}[\mathrm{Thetao}] \\ -\mathrm{Cos}[\mathrm{Phi}]\,\mathrm{Sin}[\mathrm{Psi}] + \mathrm{Cos}[\mathrm{Psi}]\,\mathrm{Sin}[\mathrm{Phi}]\,\mathrm{Sin}[\mathrm{Thetao}] & \mathrm{Cos}[\mathrm{Phi}]\,\mathrm{Cos}[\mathrm{Psi}] + \mathrm{Sin}[\mathrm{Phi}]\,\mathrm{Sin}[\mathrm{Psi}]\,\mathrm{Sin}[\mathrm{Thetao}] & \mathrm{Cos}[\mathrm{Thetao}]\,\mathrm{Sin}[\mathrm{Phi}] \\ \mathrm{Sin}[\mathrm{Phi}]\,\mathrm{Sin}[\mathrm{Psi}] + \mathrm{Cos}[\mathrm{Phi}]\,\mathrm{Cos}[\mathrm{Psi}]\,\mathrm{Sin}[\mathrm{Thetao}] & -\mathrm{Cos}[\mathrm{Psi}]\,\mathrm{Sin}[\mathrm{Phi}] + \mathrm{Cos}[\mathrm{Phi}]\,\mathrm{Sin}[\mathrm{Psi}]\,\mathrm{Sin}[\mathrm{Thetao}] & \mathrm{Cos}[\mathrm{Phi}]\,\mathrm{Cos}[\mathrm{Thetao}] \end{pmatrix}$$

This can also be expressed using quartenions

In[261]:=

$$L_{BV} := \begin{pmatrix} q0^2 + q1^2 - q2^2 - q3^2 & 2\,(q1\,q2 + q0\,q3) & 2\,(-q0\,q2 + q1\,q3) \\ 2\,(q1\,q2 - q0\,q3) & q0^2 - q1^2 + q2^2 - q3^2 & 2\,(q0\,q1 + q2\,q3) \\ 2\,(q0\,q2 + q1\,q3) & 2\,(-q0\,q1 + q2\,q3) & q0^2 - q1^2 - q2^2 + q3^2 \end{pmatrix}$$

where the quartenions can be expressed as below. The inital values of the quartenions are caluclated using these expressions

In[262]:=

$$\text{initialExpressions} = \Big\{ \{q0, \text{Cos}[\tfrac{\phi}{2}]\,\text{Cos}[\tfrac{\theta}{2}]\,\text{Cos}[\tfrac{\psi}{2}] + \text{Sin}[\tfrac{\phi}{2}]\,\text{Sin}[\tfrac{\theta}{2}]\,\text{Sin}[\tfrac{\psi}{2}]\},$$
$$\{q1, \text{Sin}[\tfrac{\phi}{2}]\,\text{Cos}[\tfrac{\theta}{2}]\,\text{Cos}[\tfrac{\psi}{2}] - \text{Cos}[\tfrac{\phi}{2}]\,\text{Sin}[\tfrac{\theta}{2}]\,\text{Sin}[\tfrac{\psi}{2}]\},$$
$$\{q2, \text{Cos}[\tfrac{\phi}{2}]\,\text{Sin}[\tfrac{\theta}{2}]\,\text{Cos}[\tfrac{\psi}{2}] + \text{Sin}[\tfrac{\phi}{2}]\,\text{Cos}[\tfrac{\theta}{2}]\,\text{Sin}[\tfrac{\psi}{2}]\},$$
$$\{q3, \text{Cos}[\tfrac{\phi}{2}]\,\text{Cos}[\tfrac{\theta}{2}]\,\text{Sin}[\tfrac{\psi}{2}] - \text{Sin}[\tfrac{\phi}{2}]\,\text{Sin}[\tfrac{\theta}{2}]\,\text{Cos}[\tfrac{\psi}{2}]\}\Big\}$$

Out[262]=

$$\Big\{\{q0, \text{Cos}[\tfrac{\text{Phi}}{2}]\,\text{Cos}[\tfrac{\text{Psi}}{2}]\,\text{Cos}[\tfrac{\text{Thetao}}{2}] + \text{Sin}[\tfrac{\text{Phi}}{2}]\,\text{Sin}[\tfrac{\text{Psi}}{2}]\,\text{Sin}[\tfrac{\text{Thetao}}{2}]\},$$
$$\{q1, \text{Cos}[\tfrac{\text{Psi}}{2}]\,\text{Cos}[\tfrac{\text{Thetao}}{2}]\,\text{Sin}[\tfrac{\text{Phi}}{2}] - \text{Cos}[\tfrac{\text{Phi}}{2}]\,\text{Sin}[\tfrac{\text{Psi}}{2}]\,\text{Sin}[\tfrac{\text{Thetao}}{2}]\},$$
$$\{q2, \text{Cos}[\tfrac{\text{Thetao}}{2}]\,\text{Sin}[\tfrac{\text{Phi}}{2}]\,\text{Sin}[\tfrac{\text{Psi}}{2}] + \text{Cos}[\tfrac{\text{Phi}}{2}]\,\text{Cos}[\tfrac{\text{Psi}}{2}]\,\text{Sin}[\tfrac{\text{Thetao}}{2}]\},$$
$$\{q3, \text{Cos}[\tfrac{\text{Phi}}{2}]\,\text{Cos}[\tfrac{\text{Thetao}}{2}]\,\text{Sin}[\tfrac{\text{Psi}}{2}] - \text{Cos}[\tfrac{\text{Psi}}{2}]\,\text{Sin}[\tfrac{\text{Phi}}{2}]\,\text{Sin}[\tfrac{\text{Thetao}}{2}]\}\Big\}$$

Conversly the total transformation matrix from body coordinates to vehicle-carried vertical frame is thus:

In[263]:=

L_{VB} := Transpose[L_{BV}]

In[264]:=

MatrixForm[L_{VB}]

Out[264]//MatrixForm=

$$\begin{pmatrix} q0^2 + q1^2 - q2^2 - q3^2 & 2\,(q1\,q2 - q0\,q3) & 2\,(q0\,q2 + q1\,q3) \\ 2\,(q1\,q2 + q0\,q3) & q0^2 - q1^2 + q2^2 - q3^2 & 2\,(-q0\,q1 + q2\,q3) \\ 2\,(-q0\,q2 + q1\,q3) & 2\,(q0\,q1 + q2\,q3) & q0^2 - q1^2 - q2^2 + q3^2 \end{pmatrix}$$

When Euler angles are used, the state vector *stateVariables* of the system is defined as:

stateVariables = {Ub, Vb, Wb, Pb, Qb, Rb, ϕ, θ, ψ, xcg, ycg, zcg};

When quartenions are used it instead becomes:

In[265]:=

```
systemVariables = {Ub,Vb,Wb,Pb,Qb,Rb,q0,q1,q2,q3,xcg,ycg,zcg};
```

In[266]:=

```
variableLowLimits = {xcg, 0.}
```

Out[266]=

```
{xcg, 0.}
```

Ω_B is from Stevensson and Lewis (1.5-2),

In[267]:=

$$\Omega_B := \begin{pmatrix} 0 & -Rb & Qb \\ Rb & 0 & -Pb \\ -Qb & Pb & 0 \end{pmatrix}$$

In[268]:=

$$\Omega_q := \begin{pmatrix} 0 & Pb & Qb & Rb \\ -Pb & 0 & -Rb & Qb \\ -Qb & Rb & 0 & -Pb \\ -Rb & -Qb & Pb & 0 \end{pmatrix}$$

In[269]:=

$$J := \begin{pmatrix} Ix & 0 & -Ixz \\ 0 & Iy & 0 \\ -Ixz & 0 & Iz \end{pmatrix}$$

The invers of the inertia matrix is

In[270]:=

```
Jinv := Simplify[Inverse[J]]
```

The speed vector in body coordinates

In[271]:=

$V_{body} := \{Ub, Vb, Wb\}$

The speed vector in earth coordinates

In[272]:=

$V_{NED} := \{s\, xcg, s\, ycg, s\, zcg\}$

The force vector in body coordinates

In[273]:=

$F_b := \{Fx, Fy, Fz\}$

The gravitational vector is defined as (earth coordinates):

In[274]:=

$G_0 := \{0, 0, g0\}$

```
Fground := {0, 0, -onPositive[zcg] kground zcg}
```

The vector of Euler angles is:

$\Phi := \{\phi, \theta, \psi\}$

The quartenion vector is:

In[276]:=

```
quart := {q0, q1, q2, q3}
```

The angular rates in body coordinates

In[277]:=

$\omega_b := \{Pb, Qb, Rb\}$

The vector of moments in body coordinates

In[278]:=

$T_b := \{Lb, Mb, Nb\}$

The system equations

Force Equations

In[279]:=

$$forceEquation := s\, V_{body} - \left(-\Omega_B . V_{body} + L_{BV} . \left(G_0 + \frac{Fground}{mass}\right) + \frac{F_b}{mass}\right)$$

Kinematic Equations

(when Euler angles are used)

$\texttt{kinematicEquation} := s\,\Phi - \zeta.\,\omega_b$

(when quartenions are used)

In[280]:=

$$\texttt{kinematicEquation} := s\;\texttt{quart} - \left(-\frac{1}{2}\,\Omega_q.\texttt{quart}\right)$$

Moment Equations

In[281]:=

$$\texttt{momentEquation} := \texttt{Simplify}[s\,\omega_b - (-\texttt{Jinv}.\Omega_B.\texttt{J}.\omega_b + \texttt{Jinv}.T_b)]$$

Naviagation Equations

In[282]:=

$$\texttt{navigationEquation} := V_{NED} - L_{VB}.V_{body}$$

Tranformation between quartenions and Euler angles can be done in a straightforward way by simply adding this set of equation to the *systemEquationDa*.

In[283]:=

$$\texttt{quartenion2Euler} := \Big\{$$
$$q1 - \left(\text{Sin}\left[\frac{\phi}{2}\right]\text{Cos}\left[\frac{\theta}{2}\right]\text{Cos}\left[\frac{\psi}{2}\right] - \text{Cos}\left[\frac{\phi}{2}\right]\text{Sin}\left[\frac{\theta}{2}\right]\text{Sin}\left[\frac{\psi}{2}\right]\right),$$
$$q2 - \left(\text{Cos}\left[\frac{\phi}{2}\right]\text{Sin}\left[\frac{\theta}{2}\right]\text{Cos}\left[\frac{\psi}{2}\right] + \text{Sin}\left[\frac{\phi}{2}\right]\text{Cos}\left[\frac{\theta}{2}\right]\text{Sin}\left[\frac{\psi}{2}\right]\right),$$
$$q3 - \left(\text{Cos}\left[\frac{\phi}{2}\right]\text{Cos}\left[\frac{\theta}{2}\right]\text{Sin}\left[\frac{\psi}{2}\right] - \text{Sin}\left[\frac{\phi}{2}\right]\text{Sin}\left[\frac{\theta}{2}\right]\text{Cos}\left[\frac{\psi}{2}\right]\right)$$
$$\Big\}$$

A (marginally) more computationally efficient way, however, is to calculate them explicitly (which is used here).

In[284]:=

$$\phi_{expr} = \text{Atan2L}\left[2\,(q2\,q3 + q0\,q1),\ q0^2 - q1^2 - q2^2 + q3^2\right];$$
$$\theta_{expr} = \text{ArcSinL}[2\,(q0\,q2 - q1\,q3)];$$
$$\psi_{expr} = \text{Atan2L}\left[2\,(q1\,q2 + q0\,q3),\ q0^2 + q1^2 - q2^2 - q3^2\right];$$

To solve the system of equations all equations are equal to zero.

Another instance of the navigation equation is needed to obtain the speeds i earth coordinates.

In[287]:=

$$\texttt{expressionVE} := \texttt{Transpose}[\{\{vx, vy, vz\}, L_{VB}.V_{body}\}]$$

Assembling the system of equations

In[288]:=

```
forceEquation // TableForm
```

Out[288]//TableForm=

$$-\frac{Fx}{mass} + s\,Ub - Rb\,Vb + Qb\,Wb - 2\,(-q0\,q2 + q1\,q3)\left(g0 - \frac{kground\;zcg\;onPositive[zcg]}{mass}\right)$$
$$-\frac{Fy}{mass} + Rb\,Ub + s\,Vb - Pb\,Wb - 2\,(q0\,q1 + q2\,q3)\left(g0 - \frac{kground\;zcg\;onPositive[zcg]}{mass}\right)$$
$$-\frac{Fz}{mass} - Qb\,Ub + Pb\,Vb + s\,Wb - \left(q0^2 - q1^2 - q2^2 + q3^2\right)\left(g0 - \frac{kground\;zcg\;onPositive[zcg]}{mass}\right)$$

In[289]:=

```
momentEquation // TableForm
```

Out[289]//TableForm=

$$\frac{-\mathrm{Iz}^2\,\mathrm{Qb}\,\mathrm{Rb}+\mathrm{Iz}\,(\mathrm{Lb}+\mathrm{Ixz}\,\mathrm{Pb}\,\mathrm{Qb}+\mathrm{Iy}\,\mathrm{Qb}\,\mathrm{Rb}-\mathrm{Ix}\,\mathrm{Pb}\,\mathrm{s})+\mathrm{Ixz}\,(\mathrm{Nb}+\mathrm{Ix}\,\mathrm{Pb}\,\mathrm{Qb}-\mathrm{Iy}\,\mathrm{Pb}\,\mathrm{Qb}-\mathrm{Ixz}\,\mathrm{Qb}\,\mathrm{Rb}+\mathrm{Ixz}\,\mathrm{Pb}\,\mathrm{s})}{\mathrm{Ixz}^2-\mathrm{Ix}\,\mathrm{Iz}}$$

$$\frac{-\mathrm{Mb}+\mathrm{Ix}\,\mathrm{Pb}\,\mathrm{Rb}-\mathrm{Iz}\,\mathrm{Pb}\,\mathrm{Rb}+\mathrm{Ixz}\,\left(\mathrm{Pb}^2-\mathrm{Rb}^2\right)+\mathrm{Iy}\,\mathrm{Qb}\,\mathrm{s}}{\mathrm{Iy}}$$

$$\frac{\mathrm{Ixz}\,(\mathrm{Lb}-(\mathrm{Ix}-\mathrm{Iy}+\mathrm{Iz})\,\mathrm{Qb}\,\mathrm{Rb})+\mathrm{Ixz}^2\,(\mathrm{Pb}\,\mathrm{Qb}+\mathrm{Rb}\,\mathrm{s})+\mathrm{Ix}\,(\mathrm{Nb}+\mathrm{Ix}\,\mathrm{Pb}\,\mathrm{Qb}-\mathrm{Iy}\,\mathrm{Pb}\,\mathrm{Qb}-\mathrm{Iz}\,\mathrm{Rb}\,\mathrm{s})}{\mathrm{Ixz}^2-\mathrm{Ix}\,\mathrm{Iz}}$$

In[290]:=

```
kinematicEquation // TableForm
```

Out[290]//TableForm=

$$\frac{1}{2}\,(\mathrm{Pb}\,\mathrm{q1}+\mathrm{q2}\,\mathrm{Qb}+\mathrm{q3}\,\mathrm{Rb})+\mathrm{q0}\,\mathrm{s}$$

$$\frac{1}{2}\,(-\mathrm{Pb}\,\mathrm{q0}+\mathrm{q3}\,\mathrm{Qb}-\mathrm{q2}\,\mathrm{Rb})+\mathrm{q1}\,\mathrm{s}$$

$$\frac{1}{2}\,(-\mathrm{Pb}\,\mathrm{q3}-\mathrm{q0}\,\mathrm{Qb}+\mathrm{q1}\,\mathrm{Rb})+\mathrm{q2}\,\mathrm{s}$$

$$\frac{1}{2}\,(\mathrm{Pb}\,\mathrm{q2}-\mathrm{q1}\,\mathrm{Qb}-\mathrm{q0}\,\mathrm{Rb})+\mathrm{q3}\,\mathrm{s}$$

In[291]:=

```
navigationEquation // TableForm
```

Out[291]//TableForm=

$$-\left(\mathrm{q0}^2+\mathrm{q1}^2-\mathrm{q2}^2-\mathrm{q3}^2\right)\mathrm{Ub}-2\,(\mathrm{q1}\,\mathrm{q2}-\mathrm{q0}\,\mathrm{q3})\,\mathrm{Vb}-2\,(\mathrm{q0}\,\mathrm{q2}+\mathrm{q1}\,\mathrm{q3})\,\mathrm{Wb}+\mathrm{s}\,\mathrm{xcg}$$

$$-2\,(\mathrm{q1}\,\mathrm{q2}+\mathrm{q0}\,\mathrm{q3})\,\mathrm{Ub}-\left(\mathrm{q0}^2-\mathrm{q1}^2+\mathrm{q2}^2-\mathrm{q3}^2\right)\mathrm{Vb}-2\,(-\mathrm{q0}\,\mathrm{q1}+\mathrm{q2}\,\mathrm{q3})\,\mathrm{Wb}+\mathrm{s}\,\mathrm{ycg}$$

$$-2\,(-\mathrm{q0}\,\mathrm{q2}+\mathrm{q1}\,\mathrm{q3})\,\mathrm{Ub}-2\,(\mathrm{q0}\,\mathrm{q1}+\mathrm{q2}\,\mathrm{q3})\,\mathrm{Vb}-\left(\mathrm{q0}^2-\mathrm{q1}^2-\mathrm{q2}^2+\mathrm{q3}^2\right)\mathrm{Wb}+\mathrm{s}\,\mathrm{zcg}$$

The assembled set of equations

In[292]:=

```
systemEquationsDa :=
 Flatten[{forceEquation, momentEquation, kinematicEquation, navigationEquation}]
```

In[293]:=

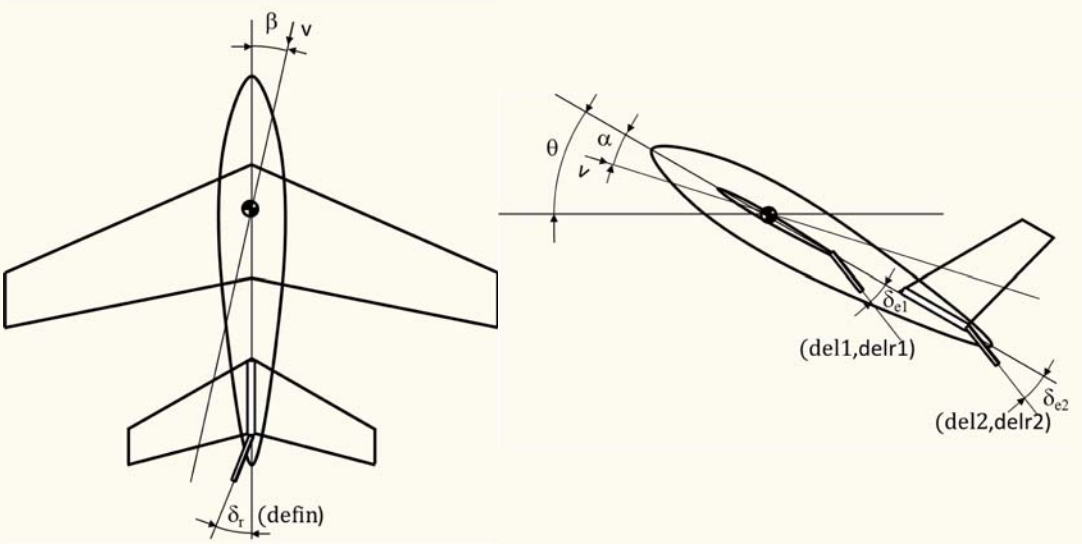
```
boundaryEquations = {};
```

In[294]:=
```
systemEquationsDa // TableForm
```

Out[294]//TableForm=

$$-\frac{Fx}{mass} + s\,Ub - Rb\,Vb + Qb\,Wb - 2\,(-q0\,q2 + q1\,q3)\,\left(g0 - \frac{kground\,zcg\,onPositive[zcg]}{mass}\right)$$
$$-\frac{Fy}{mass} + Rb\,Ub + s\,Vb - Pb\,Wb - 2\,(q0\,q1 + q2\,q3)\,\left(g0 - \frac{kground\,zcg\,onPositive[zcg]}{mass}\right)$$
$$-\frac{Fz}{mass} - Qb\,Ub + Pb\,Vb + s\,Wb - \left(q0^2 - q1^2 - q2^2 + q3^2\right)\,\left(g0 - \frac{kground\,zcg\,onPositive[zcg]}{mass}\right)$$
$$\frac{-Iz^2\,Qb\,Rb + Iz\,(Lb + Ixz\,Pb\,Qb + Iy\,Qb\,Rb - Ix\,Pb\,s) + Ixz\,(Nb + Ix\,Pb\,Qb - Iy\,Pb\,Qb - Ixz\,Qb\,Rb + Ixz\,Pb\,s)}{Ixz^2 - Ix\,Iz}$$
$$\frac{-Mb + Ix\,Pb\,Rb - Iz\,Pb\,Rb + Ixz\,\left(Pb^2 - Rb^2\right) + Iy\,Qb\,s}{Iy}$$
$$\frac{Ixz\,(Lb - (Ix - Iy + Iz)\,Qb\,Rb) + Ixz^2\,(Pb\,Qb + Rb\,s) + Ix\,(Nb + Ix\,Pb\,Qb - Iy\,Pb\,Qb - Iz\,Rb\,s)}{Ixz^2 - Ix\,Iz}$$
$$\frac{1}{2}\,(Pb\,q1 + q2\,Qb + q3\,Rb) + q0\,s$$
$$\frac{1}{2}\,(-Pb\,q0 + q3\,Qb - q2\,Rb) + q1\,s$$
$$\frac{1}{2}\,(-Pb\,q3 - q0\,Qb + q1\,Rb) + q2\,s$$
$$\frac{1}{2}\,(Pb\,q2 - q1\,Qb - q0\,Rb) + q3\,s$$
$$-\left(q0^2 + q1^2 - q2^2 - q3^2\right)\,Ub - 2\,(q1\,q2 - q0\,q3)\,Vb - 2\,(q0\,q2 + q1\,q3)\,Wb + s\,xcg$$
$$-2\,(q1\,q2 + q0\,q3)\,Ub - \left(q0^2 - q1^2 + q2^2 - q3^2\right)\,Vb - 2\,(-q0\,q1 + q2\,q3)\,Wb + s\,ycg$$
$$-2\,(-q0\,q2 + q1\,q3)\,Ub - 2\,(q0\,q1 + q2\,q3)\,Vb - \left(q0^2 - q1^2 - q2^2 + q3^2\right)\,Wb + s\,zcg$$

The Forces and Moments Acting on the Aircraft



The forces and moments acting on the aircraft are aerodynamic forces and moments and forces and moments from the engine

In[295]:=
```
aircraft`F := aircraft`aero`F + engine`F + ground`F;
aircraft`T := aircraft`aero`T + engine`T;
```

torque on x-axis

Lb := aircraft`T[[1]][[1]]

The aerodynamic forces acting on the airplane (expressed in wind coordinates) can be expressed as:

In[297]:=
```
aircraft`aero`Fw := wing1`aero`F + wing2`aero`F + fin`aero`F + body`aero`F
```

Transformed into body coordinates

In[298]:=
```
aircraft`aero`F := L_BW.aircraft`aero`Fw
```

where the transformation matrix from wind to body coordinates is:

In[299]:=
```
L_BW = L_y[α].L_z[-β];
```

In[300]:=
```
aircraft`aero`Tw := wing1`aero`T + wing2`aero`T + fin`aero`T + body`aero`T
```

Moments in body coordinates

In[301]:=
```
aircraft`aero`T := L_BW.aircraft`aero`Tw
```

In[302]:=
```
aircraft`aero`T := aircraft`aero`Tw
```

Wing

The contributions from the wing are

In[303]:=

$$\texttt{wing1`aero`F} := \begin{pmatrix} -\mathrm{Dragl1} - \mathrm{Dragr1} \\ 0 \\ -\mathrm{Liftl1} - \mathrm{Liftr1} \end{pmatrix}$$

In[304]:=

$$\texttt{wing1`aero`T} := \begin{pmatrix} -M_{\dot{\phi}}\,\mathrm{Pbe} + (\mathrm{Liftl1} - \mathrm{Liftr1})\,\mathrm{rc1} \\ (\mathrm{xbcg} - \mathrm{xw1})\,((\mathrm{Liftl1} + \mathrm{Liftr1})\,\mathrm{Cos}[\alpha] + (\mathrm{Dragl1} + \mathrm{Dragr1})\,\mathrm{Sin}[\alpha]) + \mathrm{Moment1} \\ (-\mathrm{Dragl1} + \mathrm{Dragr1})\,\mathrm{rc1} \end{pmatrix}$$

⋮

Out[305]=

⋮

Calculating the rudder hinge forces and aerodynamic stiffness

In[306]:=
```
kal1 := CLde1 S1 q lc1;
kar1 := CLde1 S1 q lc1;
kal12 := CLde12 S1 q lc12;
kar12 := CLde12 S1 q lc12;
```

Rudder forces (wave variables for connection nodes)

In[310]:=
```
exprW1 = {
    {cxal1, kal1 (del1 + α)},
    {cxar1, kar1 (der1 + α)},
    {cxal12, kal12 (del12 + α)},
    {cxar12, kar12 (der12 + α)}};
```

Horizontal tail

The contributions from the horizontal tail are

Forces

In[311]:=
```
wing2`aero`F := ( -Dragl2 - Dragr2
                  -Cy2
                  -Liftl2 - Liftr2 )
```

Moments

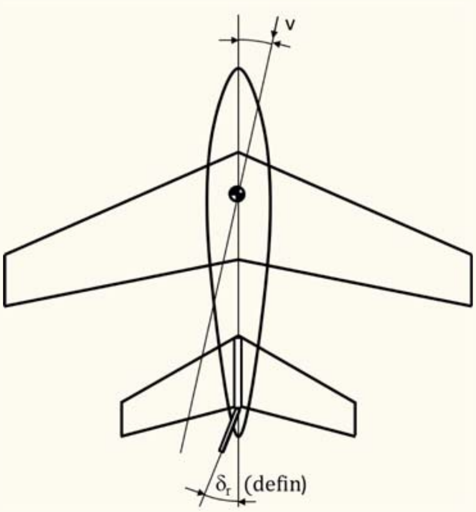
In[312]:=
```
wing2`aero`T :=
 ( (Liftl2 - Liftr2) rc2
   -M_θ̇ Qbe + ((Liftl2 + Liftr2) Cos[α] - (Dragl2 + Dragr2) Sin[α] - Cl2 - Cr2) (xbcg - xw2)
   (-Dragl2 + Dragr2) rc2 )
```

Rudder hinge forces and aerodynamic stiffness

In[313]:=
```
kal2 := CLalpha2e S2 q lc2;
kar2 := CLalpha2e S2 q lc2;
```

In[315]:=
```
exprW2 = {
   {cxal2, kal2 (del2 + α)},
   {cxar2, kar2 (der2 + α)}};
```

Fin



The contributions from the vertical tail are

Forces

In[316]:=
```
fin`aero`F := ( -Dragfin
                -Cfin
                0 )
```

Moments

In[317]:=
```
fin`aero`T := ( 0
                0
                -Cfin (xwfin - xbcg) - Rbe M_ψ̇ )
```

Rudder hinge forces and aerodynamic stiffness

In[318]:=
```
kafin := CLdefin Sfin q lcfin;
```

In[319]:=
```
exprFin = {
    {cxfin, kafin (defin + β)}
   };
```

Body

The contributions from the body are

Forces

In[320]:=
$$\texttt{body`aero`F} := \begin{pmatrix} -\texttt{Dragb} \\ -\texttt{Cbody} \\ -\texttt{Liftb} \end{pmatrix}$$

Moments

In[321]:=
$$\texttt{body`aero`T} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Propulsion

The thrust vector in body coordinates is

In[322]:=
```
engine`T := {
    yeng (thrustl Sin[dezthrustl] Cos[deythrustl] -
       thrustr Sin[dezthrustr] Cos[deythrustr]),
    - (xeng - xbcg) (thrustl Sin[dezthrustl] Cos[deythrustl] +
       thrustr Sin[dezthrustr] Cos[deythrustr]) ,
    - (xeng - xbcg) (thrustl Cos[dezthrustl] Sin[deythrustl] +
       thrustr Cos[dezthrustr] Sin[deythrustr])
   };
```

In[323]:=
```
engine`F :=
  {thrustl Cos[deythrustl] Cos[dezthrustr] + thrustr Cos[deythrustr] Cos[dezthrustr],
   -thrustl Sin[deythrustl] Cos[dezthrustr] - thrustr Sin[deythrustr] Cos[dezthrustr],
   -thrustl Cos[deythrustl] Sin[dezthrustr] - thrustr Cos[deythrustr] Sin[dezthrustr]
  }
```

The total force an moment vectors in body coordinates are then

In[324]:=

In[325]:=
$$\texttt{ground`F} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In[326]:=
$$\texttt{ground`F} := \begin{pmatrix} 0 \\ 0 \\ -\texttt{onPositive[zcg] kground zcg} \end{pmatrix}$$

Display forces and moments

The inverse of the inertia matrix

In[327]:=
```
Jinv // MatrixForm
```

Out[327]//MatrixForm=

$$\begin{pmatrix} \frac{Iz}{-Ixz^2+Ix\,Iz} & 0 & \frac{Ixz}{-Ixz^2+Ix\,Iz} \\ 0 & \frac{1}{Iy} & 0 \\ \frac{Ixz}{-Ixz^2+Ix\,Iz} & 0 & \frac{Ix}{-Ixz^2+Ix\,Iz} \end{pmatrix}$$

In[328]:=
```
wing1`aero`F // MatrixForm
```

Out[328]//MatrixForm=

$$\begin{pmatrix} -Dragl1-Dragr1 \\ 0 \\ -Liftl1-Liftr1 \end{pmatrix}$$

In[329]:=
```
wing1`aero`T // MatrixForm
```

Out[329]//MatrixForm=

$$\Bigg(qpress\,S1\,\big(-Cmde1\,thetaal1-Cmde12\,thetaal12-Cmde1\,thetaar1-Cmde12\,thetaar12-\frac{smc\,(CLde1}{}$$

In[330]:=
```
wing2`aero`F // MatrixForm
```

Out[330]//MatrixForm=

$$\begin{pmatrix} -Dragl2-Dragr2 \\ -Cydeelev\,qpress\,S2\,(thetaal2-thetaar2) \\ -Liftl2-Liftr2 \end{pmatrix}$$

In[331]:=
```
wing2`aero`T // MatrixForm
```

Out[331]//MatrixForm=

$$\Big(\begin{array}{r} (Liftl2-L \\ -Mdvtheta\,\big(Qb+2\,(q1\,q2+q0\,q3)\,wturbx+\big(q0^2-q1^2+q2^2-q3^2\big)\,wturby+2\,(-q0\,q1+q2\,q3)\,wturl \\ (-Dragl2+D \end{array}$$

In[332]:=
```
fin`aero`F // MatrixForm
```

Out[332]//MatrixForm=

$$\begin{pmatrix} -Dragfin \\ -Cfin \\ 0 \end{pmatrix}$$

In[333]:=
```
fin`aero`T // MatrixForm
```
Out[333]//MatrixForm=
```
(                                                          0
                                                            0
 -Mdvpsi (Rb + 2 (-q0 q2 + q1 q3) wturbx + 2 (q0 q1 + q2 q3) wturby + (q0^2 - q1^2 - q2^2 + q3^2) wturbz
```

In[334]:=
```
body`aero`F // MatrixForm
```
Out[334]//MatrixForm=
```
(           -Dragb
 CLalphabv qpress Sbv Sin[Beta]
            -Liftb             )
```

In[335]:=
```
body`aero`T // MatrixForm
```
Out[335]//MatrixForm=
```
(0
 0
 0)
```

In[336]:=
```
engine`F // MatrixForm
```
Out[336]//MatrixForm=
```
( thrustl Cos[deythrustl] Cos[dezthrustr] + thrustr Cos[deythrustr] Cos[dezthrustr]
 -thrustl Cos[dezthrustr] Sin[deythrustl] - thrustr Cos[dezthrustr] Sin[deythrustr]
 -thrustl Cos[deythrustl] Sin[dezthrustr] - thrustr Cos[deythrustr] Sin[dezthrustr] )
```

In[337]:=
```
engine`T // MatrixForm
```
Out[337]//MatrixForm=
```
(    yeng (thrustl Cos[deythrustl] Sin[dezthrustl] - thrustr Cos[deythrustr] Sin[dezthrus
 (xbcg - xeng) (thrustl Cos[deythrustl] Sin[dezthrustl] + thrustr Cos[deythrustr] Sin[dezt
 (xbcg - xeng) (thrustl Cos[dezthrustl] Sin[deythrustl] + thrustr Cos[dezthrustr] Sin[deyt
```

In[338]:=
```
aircraft`F // MatrixForm
```
Out[338]//MatrixForm=
```
(              (-Dragb - Dragfin - Dragl1 - Dragl2 - Dragr1 - Dragr2) Cos[Alpha] Cos[Beta] +
                                                 (-Dragb - Dragfin - Dragl1 - Dragl2 -
 (-Liftb - Liftl1 - Liftl2 - Liftr1 - Liftr2) Cos[Alpha] - kground zcg onPositive[zcg] + (-Dr
```

In[339]:=
```
aircraft`T // MatrixForm
```
Out[339]//MatrixForm=
```
(


 -Mdvtheta (Qb + 2 (q1 q2 + q0 q3) wturbx + (q0^2 - q1^2 + q2^2 - q3^2) wturby + 2 (-q0 q1 + q2 q3) wturl


```

Assembling all the rudder forces and aerodynamic stiffnesses

In[340]:=
```
expression := Flatten[{exprW1, exprW2, exprFin}, 1];
```

In[341]:=
```
expression // TableForm
```

Out[341]//TableForm=

cal1	CLde1 lc1 qpress S1 (Alpha + thetaal1)
car1	CLde1 lc1 qpress S1 (Alpha + thetaar1)
cal12	CLde12 lc12 qpress S1 (Alpha + thetaal12)
car12	CLde12 lc12 qpress S1 (Alpha + thetaar12)
cal2	CLalpha2e lc2 qpress S2 (Alpha + thetaal2)
car2	CLalpha2e lc2 qpress S2 (Alpha + thetaar2)
cfin	CLdefin lcfin qpress Sfin (Beta + thetafin)

The wing spans can be calculated as:

$b1 = \sqrt{S1\ AR1};$

and

$b2 = \sqrt{S2\ AR2};$

Roll damping term

In[342]:=

$$M_{\dot{\phi}\,expr} := q\left(S1\ CLalpha1e\ b1^2 + S2\ CLalpha2e\ b2^2\right)\left(\frac{(0.7/2)^2}{v + 0.1}\right)$$

In[343]:=

$$M_{\dot{\phi}} := q\left(S1\ CLalpha1e\ b1^2 + S2\ CLalpha2e\ b2^2\right)\left(\frac{(0.7/2)^2}{v + 0.1}\right)$$

In[344]:=

$$M1_{\dot{\phi}} := q\left(S1\ CLalpha1e\ b1^2\right)\left(\frac{(0.7/2)^2}{v + 0.1}\right)$$

In[345]:=

$$M2_{\dot{\phi}} := q\left(S2\ CLalpha2e\ b2^2\right)\left(\frac{(0.7/2)^2}{v + 0.1}\right)$$

Pitch damping term

In[346]:=

$$M_{\dot{\theta}expr} := q\ S2\ CLalpha2e\ \frac{(xw2 - xbcg)^2}{v + 0.1};$$

In[347]:=

$$M1_{\dot{\theta}expr} := q\ S1\ CLalpha2e\ \frac{(xw1 - xbcg)^2}{v + 0.1};$$

In[348]:=

$$M2_{\dot{\theta}expr} := q\ S2\ CLalpha2e\ \frac{(xw2 - xbcg)^2}{v + 0.1};$$

Yaw damping term

In[349]:=

$$M_{\dot{\psi}\,expr} := q\ Sfin\ CLalphafin\ \frac{(xwfin - xbcg)^2}{v + 0.1};$$

Calculation of aerodynamic forces

In[350]:=
```
Dragl1expr := Cdl1 S1 q;
Dragr1expr := Cdr1 S1 q;
Liftl1expr := CLl1 S1 q;
Liftr1expr := CLr1 S1 q;
Moment1 := (Cml1 + Cmr1) S1 q
Momentx1 := (((CLl1 - CLr1) Cos[alpha] + (Cdl1 - Cdr1) Sin[alpha]) rc2 +
      (CLde12 del12 - CLde12 del12 ) (rc12 - rc2)) S2 q;
Dragl2expr := Cdl2 S2 q;
Dragr2expr := Cdr2 S2 q;
Liftl2expr := CLl2 S2 q;
Liftr2expr := CLr2 S2 q;
Cl2expr := CLl2 S2 q;
Cr2expr := -CLr2 S2 q;
Momentx2 := ((CLl2 - CLr2) Cos[alpha] + (Cdl2 - Cdr2) Sin[alpha]) rc2 S2 q;
Dragfinexpr := Cd0fin Sfin q;
Cfinexpr := CLfin Sfin q;
Dragbexpr := (Cd0b Sbh + CdW S1) q;
Liftbexpr := CLbh Sbh q;
Cbody := CLbv Sbv q;
Cy2 := Cydeelev (del2 - der2) S2 q;
```

The wing drag is a combiantion of parasitic drag and induced drag

In[369]:=
```
Cdl1 = Cd01 / 2 + Cdil1;
Cdr1 = Cd01 / 2 + Cdir1;
Cdl2 = Cd02 / 2 + Cdil2;
Cdr2 = Cd02 / 2 + Cdir2;
```

The dynamic pressure is calculated as

In[373]:=

$$q_{expr} := rho \frac{v^2}{2};$$

In[374]:=
```
epsv = 0.0001;
```

In[375]:=

$$\{uw, uv, ww\} = L_{VB}.\{vturbx, vturby, vturbz\}$$

Out[375]=

$$\{(q0^2 + q1^2 - q2^2 - q3^2)\ vturbx + 2\ (q1\ q2 - q0\ q3)\ vturby + 2\ (q0\ q2 + q1\ q3)\ vturbz,\ 2\ (q1\ q2 + q0\ q3)\ vturbx + (q0^2 - q1^2 + q2^2 - q3^2)\ vturby + 2\ (-q0\ q1 + q2\ q3)\ vturbz,\ 2\ (-q0\ q2 + q1\ q3)\ vturbx + 2\ (q0\ q1 + q2\ q3)\ vturby + (q0^2 - q1^2 - q2^2 + q3^2)\ vturbz\}$$

In[376]:=

$$\{pw, qw, rw\} = L_{VB}.\{wturbx, wturby, wturbz\}$$

Out[376]=

$$\{(q0^2 + q1^2 - q2^2 - q3^2)\ wturbx + 2\ (q1\ q2 - q0\ q3)\ wturby + 2\ (q0\ q2 + q1\ q3)\ wturbz,\ 2\ (q1\ q2 + q0\ q3)\ wturbx + (q0^2 - q1^2 + q2^2 - q3^2)\ wturby + 2\ (-q0\ q1 + q2\ q3)\ wturbz,\ 2\ (-q0\ q2 + q1\ q3)\ wturbx + 2\ (q0\ q1 + q2\ q3)\ wturby + (q0^2 - q1^2 - q2^2 + q3^2)\ wturbz\}$$

In[377]:=
```
Ube = Ub + uw;
Vbe = Vb + uw;
Wbe = Wb + ww;
Pbe = Pb + pw;
Qbe = Qb + qw;
Rbe = Rb + rw;
```

In[383]:=
$$v_{expr} = \sqrt{Ube^2 + Vbe^2 + Wbe^2 + epsv};$$

α_{expr} := ArcSin[Wbe, v_{expr}];

In[384]:=
α_{expr} := Atan2L[Wbe, Ube + epsv];

In[385]:=
$$\beta_{expr} := \text{Atan2L}\left[Vbe, \sqrt{Ube^2 + Wbe^2 + epsv}\right];$$

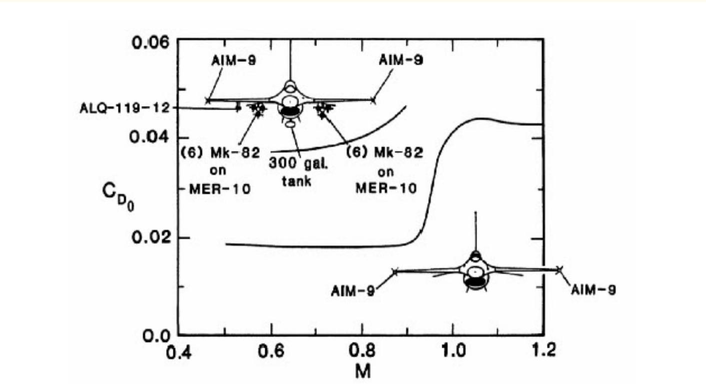
β_{expr} := ArcSin[Vb / v_{expr}]; (Etkin (4.3, 3), and Stevens and Lewis (2.3 − 6))

In[386]:=
$$\text{LogisticFunc}[x\_] := \frac{1}{1 + en^{-x}};$$

In[387]:=
$$\text{transM}[v\_, vM\_, dM\_] := \text{LogisticFunc}\left[2\,\frac{v - vM\,(1 - dM)}{vM\,dM}\right];$$

In[388]:=
```
CdW = CdW0 transM[v, vM, dM]
```

Out[388]=
$$\frac{CdW0}{1 + en^{-\frac{2\,(v-(1-dM)\,vM)}{dM\,vM}}}$$



The lifts and drag are calculated from:

In[389]:=
```
CDflat = 1.;
CDfinflat = 1.;
```

The lift coefficients. This model assumes all moving tailplane that is used on supersonic aircraft.

In[391]:=
```
CLl1 = CLift[Alpha1, CLalpha1e, ap1, an1, awp1, awn1] / 2 + CLde1 del1 + CLde12 del12;
CLr1 = CLift[Alpha1, CLalpha1e, ap1, an1, awp1, awn1] / 2 + CLde1 der1 + CLde12 der12;
CLl2 = CLift[Alpha2 + del2, CLalpha2e, ap2, an2, awp2, awn2] / 2;
CLr2 = CLift[Alpha2 + der2, CLalpha2e, ap2, an2, awp2, awn2] / 2;
Cdil1 =
  CDragInd[Alpha1, AR1, e1e, CLalpha1e, ap1, an1, awp1, awn1] / 2 + Cdide1 (del1 - de10)^2 +
   Cdide12 (del12 - de120)^2 - Cdide112 (del1 - de10) (del12 - de120);
Cdir1 = CDragInd[Alpha1, AR1, e1e, CLalpha1e, ap1, an1, awp1, awn1] / 2 + Cdide1
    (der1 - de10)^2 + Cdide12 (der12 - de120)^2 - Cdide112 (der1 - de10) (der12 - de120);
Cdil2 = CDragInd[Alpha2 + del2, AR2, e2e, CLalpha2e, ap2, an2, awp2, awn2] / 2;
Cdir2 = CDragInd[Alpha2 + der2, AR2, e2e, CLalpha2e, ap2, an2, awp2, awn2] / 2;
CLfin = CLift[-β - CLdefin defin / CLalphafin, CLalphafin, afin, afin, awfin, awfin];

Cmr1 = CMoment[Alpha1, Cm01, Cmfs1, ap1, an1, awp1, awn1] / 2 -
   Cmde1 der1 - Cmde12 der12 - CLr1 smc/4 transM[v, vM, dM];
Cml1 = CMoment[Alpha1, Cm01, Cmfs1, ap1, an1, awp1, awn1] / 2 -
   Cmde1 del1 - Cmde12 del12 - CLl1 smc/4 transM[v, vM, dM];
```

In[402]:=
```
CL1expr = (CLl1 + CLr1);
CL2expr = (CLl2 + CLr2);
Cd1expr = (Cdl1 + Cdr1);
Cd2expr = (Cdl2 + Cdr2);
CLtotexpr = (CLl1 + CLr1) + (CLl2 + CLr2) S2 / S1 + CLbh Sbh / S1 ;
CDtotexpr = (Cdl1 + Cdr1) + CdW + (Cdl2 + Cdr2) S2 / S1 + Cd0b Sbh / S1 + Cd0fin Sfin / S1;

CLbh = Sin[α] CLalphabh;
CLbv = Sin[-β] CLalphabv;

Alpha1 = α dah1 - ia1;
Alpha2 = α dah2 - ia2;
```

Weight and balance

In[412]:=
```
massexpr = Me + Mfuel + Mcargo;
```

In[413]:=
```
xbcgexpr = 1/mass (Me xbcge + Mfuel xfuel + Mcargo xcargo);
```

LocalExpressions

The geometric data is made dimmensionless using wingspan or mean aerodynamic cord mac (here derived from the standard mean cord and a dimensionless factor mac0) as reference. In this way the whole aircraft is rescaled if aspect ratio or wing reference area is changed.

In[414]:=
```
localExpressions = {
   {b1, √(S1 AR1), "m", "Wing span 1"},
   {b2, √(S2 AR2), "m", "Wing span 2"},
   {smc, √(S1 / AR1), "m", "Standard aerodynamic chord"},
```

```
{mac, mac0 smc, "m", "Mean aerodynamic chord"},
{hthrust, hthrust0 b1, double, "", "engine vert. pos"},
{Ix, Ix0 Me S1 AR1, double, " ", "Inertia moment Ix/(Me S1 AR1)"},
{Ixz, Ixz0 Me S1, double, " ", "Inertia moment"},
{Iy, Iy0 Me S1 / AR1, double, " ", "Inertia moment Iy/(Me S1/AR1)"},
{Iz, Iz0 Me S1 / AR1, double, " ", "Inertia moment Iy/(Me S1/AR1)"},
{lc1, lc10 mac, double, "", "norm. ctrl surf. 1 ac fr hinge lc1/sqrt(AR1 S1)"},
{lc2, lc20 mac, double, "", "norm. ctrl surf. 2 ac fr hinge lc1/sqrt(AR1 S1)"},
{lc12, lc120 mac, double, "", "norm. flap 1 ac fr hinge"},
{lcfin, lcfin0 mac, double, "", "ctrl s. fin ac fr hinge"},
{rc1, rc10 b1, double, "m", "norm. ctrl surface 1 mom. arm"},
{rc12, rc120 b1, double, "m", "norm. ctrl surface 12 mom. arm"},
{rc2, rc20 b1, double, "m", "norm. ctrl surface 1 mom. arm"},
{rcfin, rcfin0 mac, double, "m", "norm. ctrl surf. fin mom. arm"},
{S2, S20 S1, double, "", "norm. wing area 2"},
{Sbh, Sbh0 S1, double, "", "norm. hor. proj. area"},
{Sbv, Sbv0 S1, double, "", "norm.body vert. proj. area"},
{Sfin, Sfin0 S1, double, "", "norm. fin area"},
{xbach, xbach0 mac, double, " ", "body ac. hor."},
{xbacv, xbacv0 mac, double, " ", "body ac vert."},
{xbcge, xbcge0 mac, double, " ", "body cg"},
{xcargo, xcargo0 mac, double, " ", "cargo pos."},
{xfuel, xfuel0 mac, double, " ", ""},
{xw1, xw10 mac, double, " ", "wing1  position"},
{xw2, xw20 mac, double, " ", "wing 2 position"},
{xwfin, xwfin0 mac, double, "", ""},
{xeng, xeng0 mac, double, "m", "engines x-pos"},
{yeng, yeng0 b1, double, "m", "engines off. from center"},
{betaM, ((1 - (v/vM)^2)^2 + (epsM v/vM)^2)^(1/4), double, "Mach effect on lift"},
{CLalpha1e, CLalpha1 / betaM, double, "Effective lift sloop"},
{CLalpha2e, CLalpha2 / betaM, double, "Effective lift sloop"},
{e1e, e1 - e1 (1 - 1/AR1) transM[v, vM, dM],
 double, "Effective oswald efficieny 1"},
{e2e, e2 - e2 (1 - 1/AR2) transM[v, vM, dM], double, "Effective oswald efficieny 2"},
{v, v_expr, double, "Abs. value of speed"},
{α, α_expr, double, "Angle of attack"},
{q, q_expr, double, "Dynamic pressure"},
{β, β_expr, double, "Slip angle"},
{mass , massexpr, double, "total AC-weight"},
{xbcg, xbcgexpr, double, "AC-cg"},
{Dragl1, Dragl1expr, double, "Drag from wing 1"},
{Dragr1, Dragr1expr, double, "Drag from wing 1"},
{Liftl1, Liftl1expr, double, "Lift from wing 1"},
{Liftr1, Liftr1expr, double, "Lift from wing 1"},
{Dragl2, Dragl2expr, double, "Drag from wing 2"},
{Dragr2, Dragr2expr, double, "Drag from wing 2"},
{Liftl2, Liftl2expr, double, "Lift from wing 2"},
{Liftr2, Liftr2expr, double, "Lift from wing 2"},
{Liftb, Liftbexpr, double, "Lift from body"},
{Dragb, Dragbexpr, double, "Drag from body"},
{Cfin, Cfinexpr, double, "Force from vertical tail"},
```

```
  {Cl2, Cl2expr, double, "Side force from left canard"},
  {Cr2, Cr2expr, double, "Side force from right canard"},
  {Dragfin, Dragfinexpr, double, "Drag from body"},
  {M_θ̇, M_θ̇expr, double, "Damping term in pitch"},
  {M_ψ̇, M_ψ̇expr, double, "Damping term in yaw"},
  {Fx, aircraft`F[[1]][[1]], double, "Force in x"},
  {Fy, aircraft`F[[2]][[1]], double, "Force in y"},
  {Fz, aircraft`F[[3]][[1]], double, "Force in z"},
  {Lb, aircraft`T[[1]][[1]], double, "moment on x-axis"},
  {Mb, aircraft`T[[2]][[1]], double, "moment on y-axis"},
  {Nb, aircraft`T[[3]][[1]], double, "moment on z-axis"}};
```

In[415]:=

```
expressions = Flatten[{expression, expressionVE, {
     {AlphaAttack, α},
     {BetaSlip, β},
     {altitude, -zcg},
     {ϕ, ϕexpr},
     {θ, θexpr},
     {ψ, ψexpr},
     {gfx, Fx / mass},
     {gfy, Fy / mass},
     {gfz, Fz / mass},
     {Faz, aircraft`aero`Fw[[3]][[1]]},
     {Fax, aircraft`aero`Fw[[1]][[1]]},
     {CL1, CLtotexpr},
     {Cd1, CDtotexpr},
     {Zxfin, kafin mTimestep},
     {Zxal1, kal1 mTimestep},
     {Zxar1, kar1 mTimestep},
     {Zxal12, kal12 mTimestep},
     {Zxar12, kar12 mTimestep},
     {Zxal2, kal2 mTimestep},
     {Zxar2, kar2 mTimestep}
    }}, 1];
```

In[416]:=

```
Compgen[file]
```