PLANT Graphics

Graphics

Draw the main body as a cylinder (plate)

In[1]:=
```
vecL = 1.5;
halfHeightPlate = 0.25;
plateBase = {0, 0, -halfHeightPlate};
plateTop = {0, 0, halfHeightPlate};
plateRadius = 0.5;
armRadius  = 0.2;
armLength  = 1.25;
```

Body Plate Graphics

Main Frame

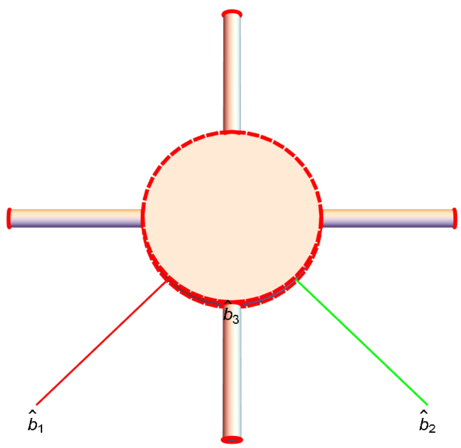
Draw the main body as a cylinder (plate)

In[8]:=
```
vecL = 1.5;
halfHeightPlate = 0.05;
plateBase = {0, 0, -halfHeightPlate};
plateTop = {0, 0, halfHeightPlate};
plateRadius = 0.5;
armRadius  = 0.05;
armLength  = 1.25;
motorPlatRadius  = 0.1;
```

In[16]:=
```
plateGraphic =
  {EdgeForm[Directive[Thick, Dashed, Red]], Cylinder[{plateBase, plateTop}, plateRadius],
   Cylinder[{{plateRadius * Sin[Pi / 4], plateRadius * Sin[Pi / 4], 0},
     {armLength * Sin[Pi / 4], armLength * Sin[Pi / 4], 0}}, armRadius],
   Cylinder[{{-plateRadius * Sin[Pi / 4], -plateRadius * Sin[Pi / 4], 0},
     {-armLength * Sin[Pi / 4], -armLength * Sin[Pi / 4], 0}}, armRadius],
   Cylinder[{{plateRadius * Sin[Pi / 4], -plateRadius * Sin[Pi / 4], 0},
     {armLength * Sin[Pi / 4], -armLength * Sin[Pi / 4], 0}}, armRadius],
   Cylinder[{{-plateRadius * Sin[Pi / 4], plateRadius * Sin[Pi / 4], 0},
     {-armLength * Sin[Pi / 4], armLength * Sin[Pi / 4], 0}}, armRadius],
   {AbsoluteThickness[2], {Text[b̂1, {vecL, 0, 0}, {0, 1}],
     Text[b̂2, {0, vecL, 0}, {0, 1}], Text[b̂3, {0, 0, -vecL}, {0, 1}],
       {AbsoluteThickness[1], RGBColor[1, 0, 0], Line[{{0, 0, 0}, {vecL, 0, 0}}]},
     {AbsoluteThickness[1], RGBColor[0, 1, 0], Line[{{0, 0, 0}, {0, vecL, 0}}]},
     {AbsoluteThickness[1], RGBColor[0, 0, 1], Line[{{0, 0, 0}, {0, 0, -vecL}}]}}}};
```

In[17]:=
```
Show[Graphics3D[plateGraphic], ViewPoint -> {1, 1, 5},
 ViewVertical -> {0, 0, 1}, ViewCenter -> {0, 0, 1}, Boxed -> False, PlotRange -> All]
```

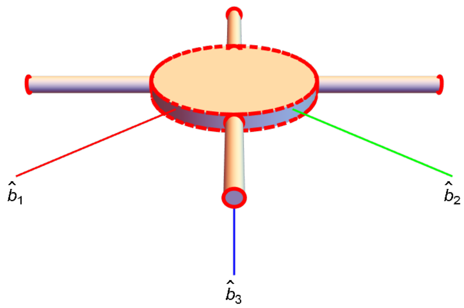
Out[17]=



In[18]:=
```
quadGraphic = {Translate[plateGraphic, {0, 0, 0}]};
```

In[19]:=
```
Show[GraphicsGrid[{{Graphics3D[quadGraphic, ViewPoint → {1, 1, 1},
      ViewVertical → {0, 0, 1}, ViewCenter → {1/2, 1/2, 1/2}, Boxed → False, PlotRange → All]}}]]
```

Out[19]=



PLANT Dynamics

Motor Mixing

Let’s assume we don’t know the rotor spinning direction and let say all of them spins in one direction.

Motor 1 (front left) and 3 rotates (rear left) clockwise.
Motor 2 (front right) and 4 rotates (rear right) counter clockwise.

In[20]:=
```
l = L / (2 * Sqrt[2]); (*L is the arm length (one rotor to another rotor along same axis),
and l is the moment arm length between x or y axis to the rotors*)
```

In[21]:=
```
κ; (*drag/thrust, with units of N.m/N*)
```

In[22]:=
```
cT; (*Total collective thrust*)
```

In[23]:=
```
δTx = momentCmdX / l (*Differential Thrust in body x-axis*)
```

Out[23]= $\frac{2\sqrt{2}\ \text{momentCmdX}}{L}$

In[24]:=
```
δTy = momentCmdY / l(*Differential Thrust in body y-axis*)
```

Out[24]= $\frac{2\sqrt{2}\ \text{momentCmdY}}{L}$

In[25]:=
```
δTz = momentCmdZ * κ(*Differential Thrust in body z-axis*)
```

Out[25]= momentCmdZ κ

```
In[26]:= eqMM1 = cT == F1 + F2 + F3 + F4;
eqMM2 = δTx == (F1 - F2 + F3 - F4);
eqMM3 = δTy == (F1 + F2 - F3 - F4 );
eqMM4 = δTz == (- F1 + F2 + F3 - F4);
(*-momentCmdZ*Kappa = r_bar*)(*moment in z-axis are just the reaction moments*)
```

```
In[30]:= MMsol = Solve[{eqMM1, eqMM2, eqMM3, eqMM4}, {F1, F2, F3, F4}] // FullSimplify
```

Out[30]=

$$\left\{\left\{F1 \to \frac{1}{4}\left(cT + \frac{2\sqrt{2}\ (\text{momentCmdX} + \text{momentCmdY})}{L} - \text{momentCmdZ}\ \kappa\right),\right.\right.$$

$$F2 \to \frac{1}{4}\left(cT + \frac{2\sqrt{2}\ (-\text{momentCmdX} + \text{momentCmdY})}{L} + \text{momentCmdZ}\ \kappa\right),$$

$$F3 \to \frac{1}{4}\left(cT + \frac{2\sqrt{2}\ (\text{momentCmdX} - \text{momentCmdY})}{L} + \text{momentCmdZ}\ \kappa\right),$$

$$\left.\left.F4 \to \frac{1}{4}\left(cT - \frac{2\sqrt{2}\ (\text{momentCmdX} + \text{momentCmdY})}{L} - \text{momentCmdZ}\ \kappa\right)\right\}\right\}$$

Here is the function created for motor mixing:
Inputs are commanded by controller: (collectThrust cT, momentCmdX, momentCmdY,momentCmdZ)
Outputs are the necessary Thrust to be produced by individual motors (F_i, i=1,2,3,4).

```
In[31]:= GenerateMotorCommands[cT_, momentCmdX_, momentCmdY_, momentCmdZ_] :=
 {1 / 4 (cT + 2 Sqrt[2] (momentCmdX + momentCmdY) / L - momentCmdZ κ),
  1 / 4 (cT + 2 Sqrt[2] (-momentCmdX + momentCmdY) / L + momentCmdZ κ),
  1 / 4 (cT + 2 Sqrt[2] (momentCmdX - momentCmdY) / L + momentCmdZ κ),
  1 / 4 (cT - 2 Sqrt[2] (momentCmdX + momentCmdY) / L - momentCmdZ κ) }
```

Euler-Newton Equations to get linear and angular accelerations

Generic Rotations for Clockwise Direction

```
In[32]:= rot1[γ_] = RotationMatrix[γ, {1, 0, 0}]

Out[32]= {{1, 0, 0}, {0, Cos[γ], -Sin[γ]}, {0, Sin[γ], Cos[γ]}}

In[33]:= MatrixForm [rot1[ϕ]]
```

Out[33]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}[\phi] & -\text{Sin}[\phi] \\ 0 & \text{Sin}[\phi] & \text{Cos}[\phi] \end{pmatrix}$$

```
In[34]:= rot2[γ_] = RotationMatrix[γ, {0, 1, 0}]

Out[34]= {{Cos[γ], 0, Sin[γ]}, {0, 1, 0}, {-Sin[γ], 0, Cos[γ]}}
```

In[35]:=
```
MatrixForm[rot2[θ]]
```

Out[35]//MatrixForm=

$$\begin{pmatrix} \text{Cos}[\theta] & 0 & \text{Sin}[\theta] \\ 0 & 1 & 0 \\ -\text{Sin}[\theta] & 0 & \text{Cos}[\theta] \end{pmatrix}$$

In[36]:=
```
rot3[γ_] = RotationMatrix[γ, {0, 0, 1}]
```

Out[36]= {{Cos[γ], -Sin[γ], 0}, {Sin[γ], Cos[γ], 0}, {0, 0, 1}}

In[37]:=
```
MatrixForm[rot3[ψ]]
```

Out[37]//MatrixForm=

$$\begin{pmatrix} \text{Cos}[\psi] & -\text{Sin}[\psi] & 0 \\ \text{Sin}[\psi] & \text{Cos}[\psi] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation Matrix or Direction Cosine Matrix

In[38]:=
```
Rot[ϕ_, θ_, ψ_] := rot3[ψ].rot2[θ].rot1[ϕ]
```

In[39]:=
```
MatrixForm[Rot[ϕ, θ, ψ]]
```

Out[39]//MatrixForm=

$$\begin{pmatrix} \text{Cos}[\theta]\,\text{Cos}[\psi] & \text{Cos}[\psi]\,\text{Sin}[\theta]\,\text{Sin}[\phi] - \text{Cos}[\phi]\,\text{Sin}[\psi] & \text{Cos}[\phi]\,\text{Cos}[\psi]\,\text{Sin}[\theta] + \text{Sin}[\phi]\,\text{Sin}[\psi] \\ \text{Cos}[\theta]\,\text{Sin}[\psi] & \text{Cos}[\phi]\,\text{Cos}[\psi] + \text{Sin}[\theta]\,\text{Sin}[\phi]\,\text{Sin}[\psi] & -\text{Cos}[\psi]\,\text{Sin}[\phi] + \text{Cos}[\phi]\,\text{Sin}[\theta]\,\text{Sin}[\psi] \\ -\text{Sin}[\theta] & \text{Cos}[\theta]\,\text{Sin}[\phi] & \text{Cos}[\theta]\,\text{Cos}[\phi] \end{pmatrix}$$

Newton’s equation in Newtonian Frame.

Fnet = m a in Newtonian Frame.

In[40]:=
```
R = {{R11, R12, R13}, {R21, R22, R23}, {R31, R32, R33}};
```

In[41]:=
```
eqF = MatrixForm[m {{accx}, {accy}, {accz}}] ==
  MatrixForm[{{0}, {0}, {m * g}} + Transpose[R].{{0}, {0}, {-Ftotal}}]
(* 3rd axid of N-E-D and gravity are both pointing
  downwards and hence 'mg' is positive*)
```

Out[41]=

$$\begin{pmatrix} m\,acc_x \\ m\,acc_y \\ m\,acc_z \end{pmatrix} == \begin{pmatrix} -R31\,F_{total} \\ -R32\,F_{total} \\ g\,m - R33\,F_{total} \end{pmatrix}$$

In[42]:=
```
MatrixForm[m {{accx}, {accy}, {accz}}] ==
 MatrixForm[{{0}, {0}, {m * g}} + Transpose[Rot[ϕ, θ, ψ]].{{0}, {0}, {-Ftotal}}]
```

Out[42]=

$$\begin{pmatrix} m\,acc_x \\ m\,acc_y \\ m\,acc_z \end{pmatrix} == \begin{pmatrix} \text{Sin}[\theta]\,F_{total} \\ -\text{Cos}[\theta]\,\text{Sin}[\phi]\,F_{total} \\ g\,m - \text{Cos}[\theta]\,\text{Cos}[\phi]\,F_{total} \end{pmatrix}$$

Euler equations in Body Frame.

In Newtonian frame, τnet =d/dt[Iω]. However, Inertia Tensor (I) is highly undesirable to measure in Newtonian frame. In body frame, I remains constant. So, we write all the moments in body frame. Doing so, we need to account for “angular velocity” of rotating frame.

τ_{net} = Iω' + ω x (Iω)

In[43]:=
```
eqM = MatrixForm[{{Ib11 pdot}, {Ib22 qdot}, {Ib33 rdot}}] ==
  MatrixForm[{{τx - (Ib33 - Ib22) q r}, {τy - (Ib11 - Ib33) p r}, {τz - (Ib22 - Ib11) p q}}]
```

Out[43]=
$$\begin{pmatrix} Ib_{11}\, p_{dot} \\ Ib_{22}\, q_{dot} \\ Ib_{33}\, r_{dot} \end{pmatrix} == \begin{pmatrix} -q\, r\, (-Ib_{22} + Ib_{33}) + \tau_x \\ -p\, r\, (Ib_{11} - Ib_{33}) + \tau_y \\ -p\, q\, (-Ib_{11} + Ib_{22}) + \tau_z \end{pmatrix}$$

Body Rates to Euler Rates

In[44]:=
```
eqω = MatrixForm[{{p}, {q}, {r}}] == MatrixForm[
   rot1[ϕ].rot2[θ].{{0}, {0}, {ψdot}} + rot1[ϕ].{{0}, {θdot}, {0}} + {{ϕdot}, {0}, {0}}]
```

Out[44]=
$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} == \begin{pmatrix} \phi_{dot} + \mathrm{Sin}[\theta]\, \psi_{dot} \\ \mathrm{Cos}[\phi]\, \theta_{dot} - \mathrm{Cos}[\theta]\, \mathrm{Sin}[\phi]\, \psi_{dot} \\ \mathrm{Sin}[\phi]\, \theta_{dot} + \mathrm{Cos}[\theta]\, \mathrm{Cos}[\phi]\, \psi_{dot} \end{pmatrix}$$

In[45]:=
```
eqEulerdot = MatrixForm[{{ϕdot}, {θdot}, {ψdot}}] == MatrixForm[
    Inverse[{{1, 0, Sin[θ]}, {0, Cos[ϕ], -Cos[θ] Sin[ϕ]}, {0, Sin[ϕ], Cos[θ] Cos[ϕ]}}].
     {{p}, {q}, {r}}] // Simplify
```

Out[45]=
$$\begin{pmatrix} \phi_{dot} \\ \theta_{dot} \\ \psi_{dot} \end{pmatrix} == \begin{pmatrix} p - r\, \mathrm{Cos}[\phi]\, \mathrm{Tan}[\theta] + q\, \mathrm{Sin}[\phi]\, \mathrm{Tan}[\theta] \\ q\, \mathrm{Cos}[\phi] + r\, \mathrm{Sin}[\phi] \\ \mathrm{Sec}[\theta]\, (r\, \mathrm{Cos}[\phi] - q\, \mathrm{Sin}[\phi]) \end{pmatrix}$$

PLANT Controller

Body Rate Controller

Inputs: pqrCmd -> desired , pqr-> measured.

Outputs: momentCmd -> τ_x, τ_y,τ_z

In[46]:=
```
bodyRateErr = pqrCmd - pqr;
omegadotCmd = bodyRateErr * kpPQR;
momentCmd = Ibb * omegadotCmd;
```

Roll Pitch Controller

Inputs: accelCmd in global XY coordinates, Rotation or DCM matrix, desired collect thrust command (cT) which is in Newtons.

Outputs: pCmd, qCmd.

rCmd is left to be zero.

In[49]:=
```
Rotmat = Rot[ϕ[t], θ[t], ψ[t]]
```
Out[49]=
```
{{Cos[θ[t]] Cos[ψ[t]], Cos[ψ[t]] Sin[θ[t]] Sin[ϕ[t]] - Cos[ϕ[t]] Sin[ψ[t]],
  Cos[ϕ[t]] Cos[ψ[t]] Sin[θ[t]] + Sin[ϕ[t]] Sin[ψ[t]]},
 {Cos[θ[t]] Sin[ψ[t]], Cos[ϕ[t]] Cos[ψ[t]] + Sin[θ[t]] Sin[ϕ[t]] Sin[ψ[t]],
  -Cos[ψ[t]] Sin[ϕ[t]] + Cos[ϕ[t]] Sin[θ[t]] Sin[ψ[t]]},
 {-Sin[θ[t]], Cos[θ[t]] Sin[ϕ[t]], Cos[θ[t]] Cos[ϕ[t]]}}
```

In[50]:=
```
MatrixForm[Rotmat]
```
Out[50]//MatrixForm=
```
( Cos[θ[t]] Cos[ψ[t]]  Cos[ψ[t]] Sin[θ[t]] Sin[ϕ[t]] - Cos[ϕ[t]] Sin[ψ[t]]   Cos[ϕ[t]] Cos[ψ[
  Cos[θ[t]] Sin[ψ[t]]  Cos[ϕ[t]] Cos[ψ[t]] + Sin[θ[t]] Sin[ϕ[t]] Sin[ψ[t]]  -Cos[ψ[t]] Sin[ϕ
     -Sin[θ[t]]                     Cos[θ[t]] Sin[ϕ[t]]                                  C
```

In[51]:=
```
ωmat = Transpose[Rotmat].Dt[Rotmat, t] // Simplify
```
Out[51]=
```
{{0, Sin[ϕ[t]] θ'[t] - Cos[θ[t]] Cos[ϕ[t]] ψ'[t],
  Cos[ϕ[t]] θ'[t] + Cos[θ[t]] Sin[ϕ[t]] ψ'[t]},
 {-Sin[ϕ[t]] θ'[t] + Cos[θ[t]] Cos[ϕ[t]] ψ'[t], 0, -ϕ'[t] + Sin[θ[t]] ψ'[t]},
 {-Cos[ϕ[t]] θ'[t] - Cos[θ[t]] Sin[ϕ[t]] ψ'[t], ϕ'[t] - Sin[θ[t]] ψ'[t], 0}}
```

In[52]:=
```
MatrixForm[ωmat]
```
Out[52]//MatrixForm=
```
(                    0                          Sin[ϕ[t]] θ'[t] - Cos[θ[t]] Cos[ϕ[t]] ψ'[t]  Cos
  -Sin[ϕ[t]] θ'[t] + Cos[θ[t]] Cos[ϕ[t]] ψ'[t]                     0
  -Cos[ϕ[t]] θ'[t] - Cos[θ[t]] Sin[ϕ[t]] ψ'[t]          ϕ'[t] - Sin[θ[t]] ψ'[t]
```

Transpose[R]. d/dt (R) = ω

In[53]:=
```
Rotmatdot = Rotmat.ωmat // Simplify;
```

From Poisson's Kinematic Equations,

R_dot = R.ω

In[54]:=
```
eq1 = R13dot == -R12 p + R11 q
```
Out[54]=
```
R13dot == q R11 - p R12
```

In[55]:=
```
eq2 = R23dot == -R22 p + R21 q
```
Out[55]=
```
R23dot == q R21 - p R22
```

Extracting upper 2X2 matrix as we only wish to find p,q for both rates.

In[56]:=
```
rmat = {{-R12, R11}, {-R22, R21}};
```

The problem looks like: $\begin{pmatrix} \text{R13dot} \\ \text{R23dot} \end{pmatrix} = \begin{pmatrix} -\text{R12} & \text{R11} \\ -\text{R22} & \text{R21} \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}$. To find $\begin{pmatrix} p \\ q \end{pmatrix}$, we take Inverse of rmat,

In[57]:=
```
eqRdot = {{pcmd}, {qcmd}} == MatrixForm[Inverse[rmat].{{R13dot}, {R23dot}}]
```
Out[57]=
$$\{\{p_{cmd}\}, \{q_{cmd}\}\} == \begin{pmatrix} \frac{\text{R13dot R21}}{-\text{R12 R21}+\text{R11 R22}} - \frac{\text{R11 R23dot}}{-\text{R12 R21}+\text{R11 R22}} \\ \frac{\text{R13dot R22}}{-\text{R12 R21}+\text{R11 R22}} - \frac{\text{R12 R23dot}}{-\text{R12 R21}+\text{R11 R22}} \end{pmatrix}$$

If we combine the Denominator or RHS matrix, which is common to all the elements using actual Rotation or DCM matrix elements,

In[58]:=
```
Rotmat[[1]][[1]] * Rotmat[[2]][[2]] - Rotmat[[1]][[2]] * Rotmat[[2]][[1]] // Simplify
```

Out[58]= $\mathrm{Cos}[\theta[t]]\ \mathrm{Cos}[\phi[t]]$

In[59]:=
```
Rotmat[[3]][[3]]
```

Out[59]= $\mathrm{Cos}[\theta[t]]\ \mathrm{Cos}[\phi[t]]$

We can replace the Denominator in eqRdot by R33.

In[60]:=
```
eqRdot /. {-R12 R21 + R11 R22 → R33}
```

Out[60]= $$\{\{p_{cmd}\},\{q_{cmd}\}\} == \begin{pmatrix} \frac{\text{R13dot R21}}{\text{R33}} - \frac{\text{R11 R23dot}}{\text{R33}} \\ \frac{\text{R13dot R22}}{\text{R33}} - \frac{\text{R12 R23dot}}{\text{R33}} \end{pmatrix}$$

The terms R13dot and R12dot are,

In[61]:=
```
R13dot = kpBank (R13cmd - R13) / τrp
    (*τrp is time constant of P-body rate controller,kpBank is the gain.*)
```

Out[61]= $$\frac{\text{kpBank}\ (-\text{R13} + \text{R13}_{cmd})}{\tau_{rp}}$$

In[62]:=
```
R23dot = kpBank (R23cmd - R23) / τrp
```

Out[62]= $$\frac{\text{kpBank}\ (-\text{R23} + \text{R23}_{cmd})}{\tau_{rp}}$$

R13_{cmd} and R23_{cmd} are obtained from following equation;

In[63]:=
```
eqF
```

Out[63]= $$\begin{pmatrix} m\ acc_x \\ m\ acc_y \\ m\ acc_z \end{pmatrix} == \begin{pmatrix} -\text{R31}\ F_{total} \\ -\text{R32}\ F_{total} \\ g\ m - \text{R33}\ F_{total} \end{pmatrix}$$

With F_{total} and accelCmd as inputs, we can solve for R13 and R23 which are the commanded elements by the controller.