Applying Vector Calculus to an Arduino RC Aircraft Autopilot System.

Define A to be a point (x, y, z) to be the airplane's current position, $\vec{v}$ to be the velocity vector of the aircraft and $\vec{g}$ to be the $A - \vec{v}$. Note that everything is in $\mathbf{R}^3$.

Left and Right Turns of the Rudder.
To determine left and right turns, first you must find the angle between $\vec{v}$ and $\vec{g}$ in the x-y plane. To do so, find

$$\theta_1 = \arccos(\frac{\vec{v} \cdot \vec{g}}{\|\vec{g}\| \cdot \|\vec{v}\|})$$

If $\theta_1 \leq \pm 5°$, we will say that our orientation is close enough to the goal orienation.
However, if $\theta_1 > \pm 5°$ the airplane must be turned in the x-y plane. You don't know the correct direction to turn. In order to find this, the simplest way to do so (at least programmatically) is to simulate turning an arbitrary direction and see if the angle increases or decreases. To rotate the direction vector, perform the computation with any θ where $\theta < 5°$ and $\theta > 0°$.

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

This simulates a rotation counter clockwise, or, a right turn. Perform the previous computation to find θ_2, the angle between $\vec{v}$ and $\vec{g}$. If $\theta_2 > \theta_1$, turn left. Otherwise, turn right. For simplicity sake, just turn the rudder the same number of degrees as θ_1. This creates an automatic PID loop.

Vertical Turns of the Elevator.
Since we cannot simplify our system to 2-D as we did for **Left and Right Turns of the Rudder**, we must be a bit more clever as we must work in 3-D. First, translate everything by $\vec{g}$. Our new location will be

$$A = (x - g_1, y - g_2, z - g_3)$$

Next, create a plane P from our new A and $\langle 1, 0, 0 \rangle$ by computing

$$A \times \langle 1, 0, 0 \rangle \cdot (x, y, z) = 0$$

$$\langle 0, -z + g_3, -y + g_2 \rangle \cdot (x, y, z) = 0$$

Next, plug in $\vec{v}$ into P. If the result is greater than 0, then the airplane must go down, if the result is less than 0 go up.

To know what angle to go upwards or downwards by, we need to project the vector $\vec{v}'$ onto P, then find the angle between our new vector and $\vec{v}$. First, take the dot product of $\vec{v}$ with the unit normal vector:

$$d = 0 + \frac{v_2 \cdot (-z + g_3)}{\sqrt{(y - g_2)^2 + (z - g_3)^2}} + \frac{v_3 \cdot (-y + g_2)}{\sqrt{(y - g_2)^2 + (z - g_3)^2}}$$

Second, project the vector onto the plane

$$\vec{r} = \vec{v} - d \cdot \langle 0, -z + g_3, -y + g_2 \rangle$$

Finally, find the angle θ_e between $\vec{r}$ and $\vec{v}$

$$\theta_e = \arccos(\frac{\vec{r} \cdot \vec{v}}{\|\vec{r}\| \cdot \|\vec{v}\|})$$

If $\theta_e \leq \pm 5°$, we will say that our orientation is close enough to the goal orienation.
However, if $\theta_e > \pm 5°$ the airplane must go up or down.

Conclusion.
By applying principles from vector calculus, we can write simple logic for an RC autopilot system. This logic is applyed in the following code: $https : //github.com/rnucuta/RCAutopilot/blob/master/RCAutopilot.ino$