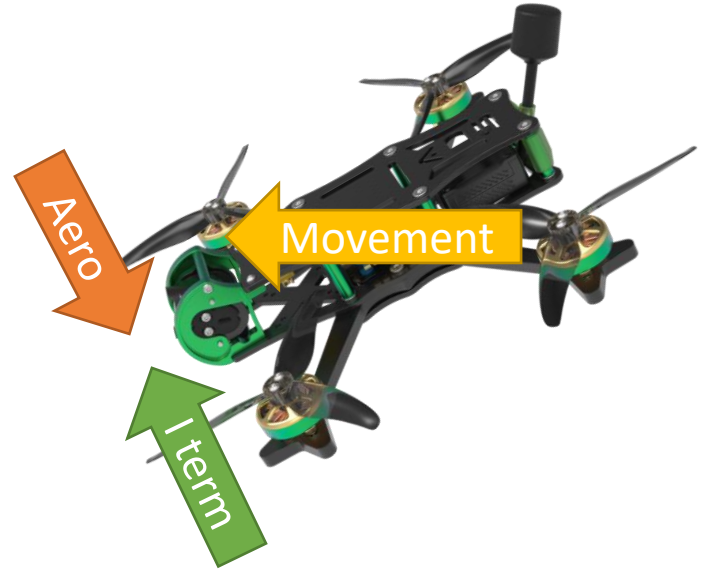
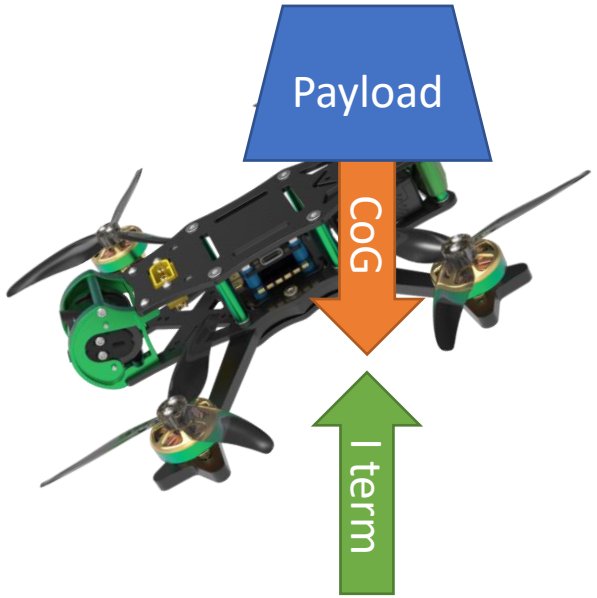
Improving I-Term Rotation

With some pizza box aerodynamics!

Role of I term on Pitch/Roll

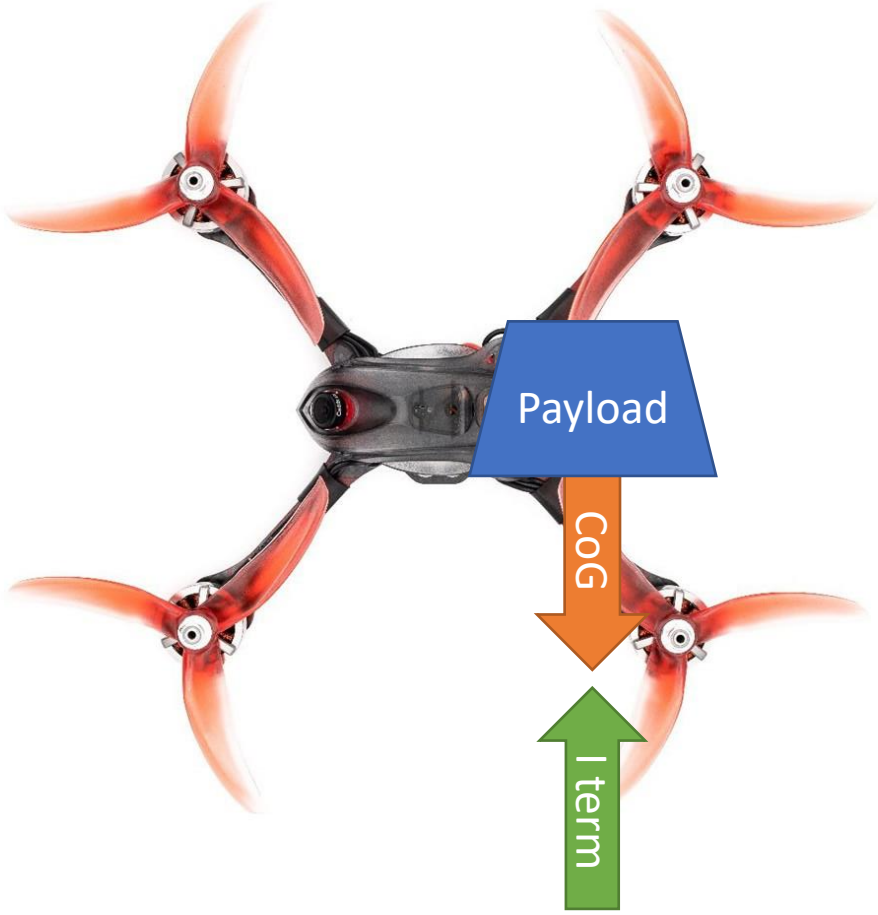
- On roll and pitch I term is primarily responsible for counteracting aerodynamic and gravity forces
- Aerodynamics and off centre CoG create forces and moments that try to rotate the quad
- The I term winds up and counteracts these forces with motor thrust
- Setpoint changes are mostly handled by P, D and FF.





Roll of I term on Yaw

- The roll of I term of Yaw is slightly different
- There are some situations where unbalanced gravity forces will need to be countered (e.g. when rolled 90deg) but these are rare
- Aerodynamic perturbations on Yaw are negligible
- Mostly I term on Yaw is supporting the P and FF terms with setpoint tracking



Why aero forces on Yaw are small

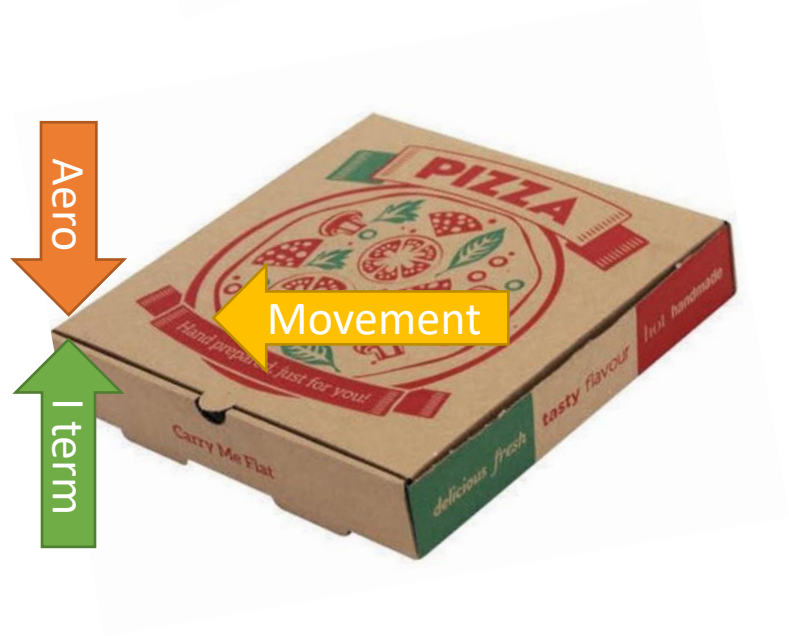
- Aerodynamically a typical quad can be thought of as similar to a pizza box
- The flat surfaces of the prop disc make the drone susceptible to being flipped on pitch and roll
- However, aero forces do not typically create unbalanced forces on Yaw due the symmetry of the shape

=

How does this affect I Term rotation?

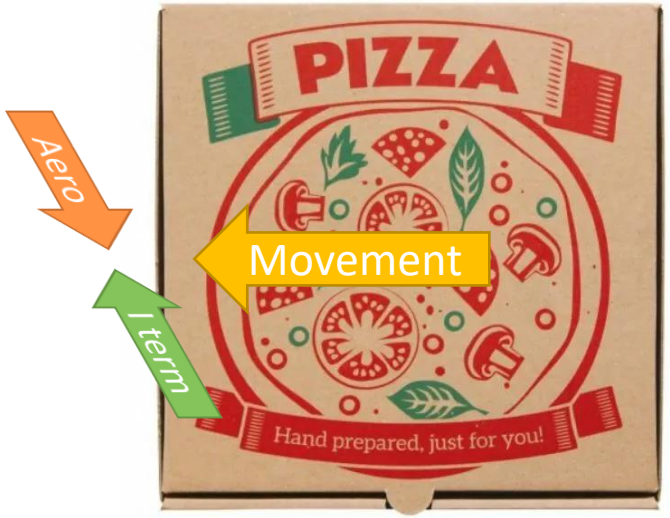
Case 1

- A pizza box is flying forwards/sideways with an I term balancing the aero forces
- The box suddenly yaws 45deg
- The aero forces don't really move (a pizza box at 45degrees looks pretty similar to one at 0deg from an aero perspective)
- Therefore from the pizza box's perspective the I term vector has rotated 45deg around yaw
- **I terms on Pitch and Roll should be rotated during a Yaw move**



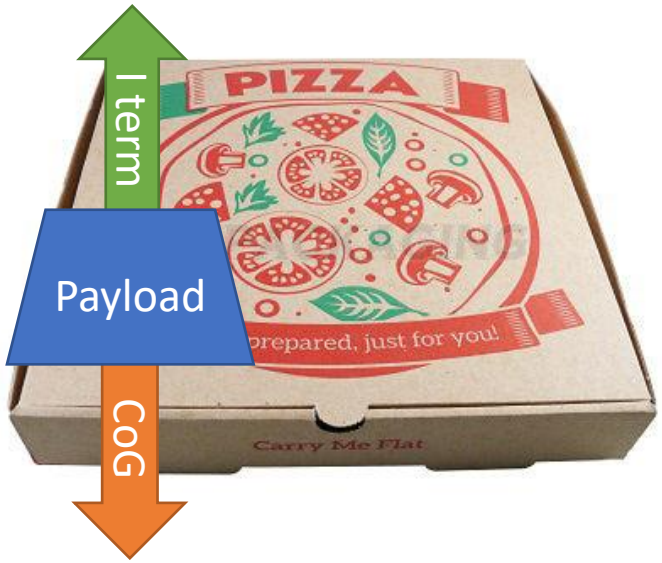
Case 2

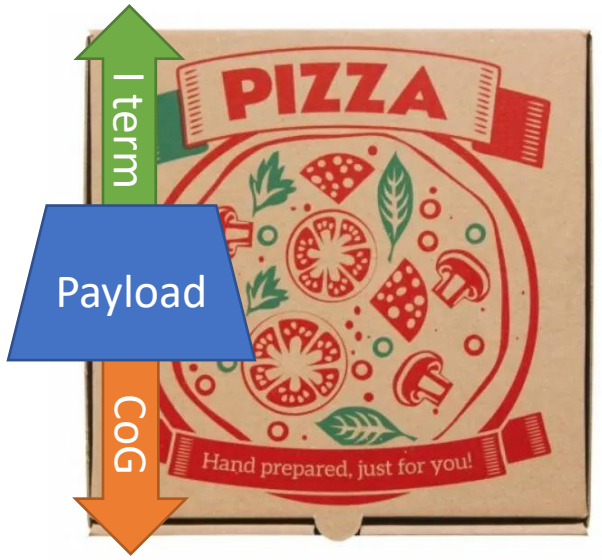
- A pizza box is flying forwards with an I term balancing the aero forces
- The box suddenly rolls 90deg
- The aero forces move immediately!
- From the pizza box's perspective the I term vector has not rotated
- **I term on Pitch should not be rotated during a Roll move**
- **I term on Roll should not be rotated during a Pitch move**



Case 3

- A pizza box is hovering with an I term balancing an off centre mass
- The box suddenly rolls 90deg
- The gravity forces move immediately!
- From the pizza box's perspective the I term vector has rotated
- However because I on Yaw and Roll/Pitch are so different it is probably not sensible to try rotate the I term in this case





Conclusion

1. It is sensible to rotate the I-Term vector between roll and pitch using the Yaw rate
    1. In most cases the forces the I term balances will move to the new axis
    2. It is easy to scale the I-Term vector (we could use the ratio of P terms on the two axes but even this is probably not necessary)
2. It is not sensible to try to rotate the I term from Yaw onto Roll/Pitch or vice versa
    1. In most cases the forces that the I term is balancing will not be acting on the new axis
    2. They are very different axes from a control perspective and it is not clear how to scale the I term during rotation

Simplified I Term rotation

src\main\flight\pid.c

```
594 594
595 595     static void rotateVector(float v[XYZ_AXIS_COUNT], float rotation[XYZ_AXIS_COUNT])
596 596     {
597 597         // rotate v around rotation vector rotation
598 598         // rotation in radians, all elements must be small
599     -       for (int i = 0; i < XYZ_AXIS_COUNT; i++) {
    599 +       for (int i = FD_YAW) {
600 600             int i_1 = (i + 1) % 3;
601 601             int i_2 = (i + 2) % 3;
602 602             float newV = v[i_1] + v[i_2] * rotation[i];
603 603             v[i_2] -= v[i_1] * rotation[i];
604 604             v[i_1] = newV;
605 605         }
606 606     }
```