Why Math Matters

The space in which a Folded Geometry flight Code ("FGC") -enabled drone calculates is unified across all sensors and systems, using a geometric algebraic space with multidimensional operators. This allows the drone to have a consistent, local predictive model of the world derived from various state estimations, known as a "Theory of World." By contrast, all other drones today—possibly excepting those in classified military programs—have a non-unified approach. Such an approach must consider:

- six dimensions (x, y, z, roll, pitch and yaw) of the drone view
- six dimensions of the world view
- similar sets of dimensions for each vision system perspective

We believe that this non-unified approach, even if technically achievable, is bug-prone and far from optimal.

Our approach—a single multi-dimensional model—maintains all perspectives at all times, and further enables vision and flight-integrated AI behaviors. We manage from 3 to 9 different multidimensional vision system inputs[1].

Our technology is implemented by performing the optimization for a particular sensor or actuator as late as possible, rather than trying to build a unified space from overly optimized, small, individual spaces. This requires a complete shift in how the software and math is applied.

Because we use the same computational space everywhere, all local viewpoints are connected by transforms. The information from one viewpoint can always be transformed to any other viewpoint.[2] Errors and uncertainties also transform cleanly, including the uncertainty in the transform itself.[3]

For example, information from gyroscopes can be mapped into the same geometric algebraic space used by an optical flow system, and vice versa. More precisely, information from gyroscopes and an imaging system can be easily fused in an information filter, such as a Kalman filter. If the drone had only these two sensors, then the orientation model—along with the full state vector, covariance matrix, and its history—would be the drone's Theory of World.

Areas of particular interest include:

- finding and correcting sensor drifts;
- internal image mapping or projection;

- vision based AI behaviors; and
- movement planning.

For drifts, an example would be correcting the true down vector. But a true down vector cannot be acquired for any flying vehicle from a purely inertial system—think of the problems with coordinated turns in planes. Some external reference is always needed. A pressure sensor can provide a reasonably good measurement of height over medium timescales (~10 sec). This is too slow for the stabilization loops running at >100Hz, but the drone's Theory of World can find a delta between the measurement and a high-quality prediction. As long as the delta is due to an error in orientation, it can compute a low-frequency correction to the down vector and keep the drone stable indefinitely.

Unlike a car or plane, a drone can rapidly move in several different directions at the same time. To use a gaming term, it can strafe right, while pitching up and down, while circling—all simultaneously. This can severely impair its imaging system; it is likely to suffer motion blur and orientation loss unless it is perfectly gimbaled, and even then, it might not be able to keep up. In a unified space, however, images can be internally projected using information from the avionics. This gives the image processing system a head start, even in situations where motion blur would otherwise be overwhelming. Importantly, information from the image processing system is also fed into avionics (akin to an optical flow sensor) because it is all part of the same Theory of World. This also allows the common technique of controlling a gimbal from the internal state to be done easily.

This approach is also important when multiple drones are used, each with their own spaces, that need to be connected. Again, by using a proper geometric space, that connection is a simple (if uncertain) transform.

The control space is also the same. Paths, orbits, or waypoints are mapped into the full geometry. For example, a GPS waypoint (latitude, longitude, and elevation) becomes a sphere in the model space, as a geometric object. Orbits or arcs also have simple geometric representations, "tubes" connecting the waypoint spheres.

And since this is the same space as the photogrammetry being done by the imaging system, it is straightforward to look for any intersections, which are likely to represent collision hazards.

Errors and uncertainties also have geometric representations, and can be handled dynamically. A location in space is represented as a sphere whose radius is either the measurement uncertainty or the required accuracy. When transformed, the uncertainty in the transform is included and the resultant radius is either larger (for a more uncertain measurement) or smaller (more accuracy required to offset the transform uncertainty). This is how the drone's Theory of World includes a concept of uncertainty.

This concept becomes particularly relevant in inspection applications, where it is important to get close enough to the target without hitting it; winds, updrafts, or turbulence around a bridge, for example, create a very dynamic uncertainty environment. Think of a sphere around the drone representing its knowledge of where it is, and a wide tube representing the path it should fly and remain within; this is a view of its representation in a mathematical expression. Should a simple intersection test fail, the drone can immediately execute a safety behavior.

As a creative experiment, consider a more “blue sky” cluster application, perhaps a dozen mapping drones in a swarm over an urban area, where the drones are constantly moving to keep even coverage over a relevant area—while not hovering in their own jet wash. Each drone feeds ephemeris information to the swarm, along with selected video via wireless links, and uses this to maintain the swarm and to do inter-drone photogrammetry on the fly. This allows both construction of a shared real-time 3D model landscape and high-accuracy relative positioning of the drones. Ground nodes, with more compute power, can texture this model with projected video to provide full “3D video” for remote-reality applications, or track objects for augmented-reality applications.

Again, the computational space is unified, with each drone now behaving like a node in a Beowulf-like cluster. The software is constructed as if the cameras and model construction are done in a single application, and then projected onto the drones, while the data flows the other way, from many drones up to the global model. Using a single space simplifies reasoning about the whole system, reduces latency, and tracks uncertainties.

In conclusion, Digital Aerolus has constructed not only a set of complementary technologies, but a consistent mathematical and system-wide underpinning, to solve both today’s flight-stability problems and the vision and AI behaviors being demanded to grow the UAS market.

[1] Counting dimensions must distinguish between dimensions and degrees of freedom. For example, FGC manages the 6 dimensions of the drone position and orientation, plus their 6 derivatives in their respective tangent spaces, which change for velocity and angular velocity, plus 12 second derivatives for acceleration, thrust, angular acceleration and torque, and that’s before they get moved around.

[2] Here we are talking about understanding information. Trying to express that every component of the system is on the same page with respect to how things are calculated as well as things like units and scales and timing—because of that moving from one view to another is a simple process (even if it has many bits) and importantly that that process is a deterministic, well understood, low risk mathematical objects—not custom-written pieces of code, written by different people with different goals for their sensors or readouts.

[3] As a simple example, think of the set of books and accounting for a project, sitting inside the set of books and accounting for a division, sitting inside the set of books and accounting for a company. If every project does things differently, it becomes harder and harder to reconcile everything at higher levels—then think of two companies merging and trying to reconcile their systems.