AEROSPACE SIMULATION

Whenever one process is represented by another, a simulation is in progress. A terminology developed recently by the military includes three categories of simulation: live, constructive, and virtual. In live simulation, actual equipment is operated by live crews. Practicing engine out procedures in an airplane with good engines or training instrument procedures while flying in good weather are live simulations. So are war game exercises played with aircraft and tanks.

Constructive simulation replaces both equipment and crews by symbols. The classical sand-table exercises, where tokens represented military units, were constructive simulations. The sand table has been computerized. It now approximates the mechanics of vehicles and even the cognitive processes of troops. Computer representations of processes ranging from water management to bacterial growth to hypersonic flow are all constructive simulations.

Virtual simulation employs live players in a simulated environment. There are still other simulations in which inanimate objects, for example, engines, sensors, control systems, or even entire missiles or unmanned aircraft are operated and tested in a virtual environment.

This article addresses virtual simulation, as it applies to the flight crews of aerospace vehicles. Regardless of the purpose of the simulation, the subject is the techniques for creating an effective virtual environment for the human pilot.

Simulators are widely used for training. Complete pilot training in a virtual simulator is not practical. A simulator suitable for this purpose, classified by the Federal Aviation Administration (FAA) as *level D,* is much more expensive than a trainer aircraft. Level D simulators are produced only for very expensive aircraft and are used for, among other things, transition training of airline pilots to new types of airliners.

On the other hand, supplementary use of simulators in flight training has long proved useful. Training pilots to fly by reference to instruments only has been accomplished since World War II by combining flight time with simulator time.

Simulators offer some unique training advantages:

- Reduction of risk.
- Reduced environmental impact.
- *Saving of Time.* The simulation can be limited to the maneuver being trained. There is no need to perform a preflight check of an aircraft, go through engine start procedure, taxi to the runway, and fly to the practice area before training can begin. No time is wasted on returning, landing, and taxiing back after the flight. The simulator can be reset to repeat a maneuver. For instance, when training landing approaches, the simulator can be reset after each approach, putting it in a position to start another approach. In live training the airplane must be flown around to the initial position, which may take anywhere from 3 min. to 15 min.
- *Control of Weather.* No time is lost due to bad weather. Yet adverse weather conditions can be conjured on demand.
- *Training Analysis.* The simulator can be "frozen" for a discussion between trainee and instructor, then continue to fly from that position.
- *Repeatability.* Flight histories can be recorded and replayed.

Beyond individual and crew training, the military uses virtual simulation for collective training. Entire units are trained while both sides of a battle are simulated. Collective training is accomplished by a technology known as *distributed interactive simulation* (DIS), which involves communications between large numbers of virtual simulators located at separate sites. Each simulator includes in the virtual environment it creates the vehicles represented by other simulators. The ultimate goal is a virtual battlefield on which live, vir-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

tual, and constructive simulations can interact. The advantages of DIS (exploited already in the Gulf War of 1991) include:

- *Cost.* In collective training, modest simulators replace expensive vehicles. There is additional savings in logistics.
- *Environmental Impact.* Live exercises tend to tear up the environment, damage property, and cause loss of life.
- *Secrecy.* Movement of units, which the enemy is likely to detect, is avoided.
- *Mission Rehearsal.* An attack can be rehearsed in the actual site while that site is still in enemy hands.
- *Debriefing.* The mission can be replayed in simulation for analysis and lessons learned.

Potential civilian use of DIS has been identified in the area of air traffic control.

The advantage of training in places and tasks that are not yet within reach is not limited to military mission rehearsal. The astronauts practiced the lunar landing in a simulator before getting a chance to perform it live for the first time.

Quite apart from training, simulation is a powerful engineering tool. Tentative designs of new flight vehicles are evaluated by experienced test pilots in virtual simulation. Much of the iterative design process by trial and error can take place in an *engineering simulator* before a prototype is built.

PRINCIPLES AND LIMITATIONS

A pilot manipulates the flight controls in response to sensory perceptions. A virtual simulator replicates the physical stimuli that induce the sensory perceptions. These perceptions, or cues, fall into several categories:

- Instrument indications.
- *Visual.* This refers to cues obtained by looking outside the vehicle. Instrument indications, even though observed visually, are addressed separately.
- *Motion.* This refers to sensations due to the pilot being moved bodily. Visual indications of motion are included in the category of visual cues.
- *Tactile.* Cues induced by the feel of the flight controls.
- *Auditory.* Cues inferred from the sound of the engine, of the airflow, and other sources.

The methods of simulating sound in a virtual simulator are no different from the ones used in reproducing music and will be discussed no further. There is an arbitrary demarcation between sound and vibration, the latter being considered a motion cue.

At some future time, the technology may be available to induce sensory perceptions by a direct link to the subject's nervous system. The present article addresses only the creation of perceptions by the normal use of the subject's sensory organs. Visual cues are produced by displays presented to the pilot; motion cues are created by moving the simulator cab.

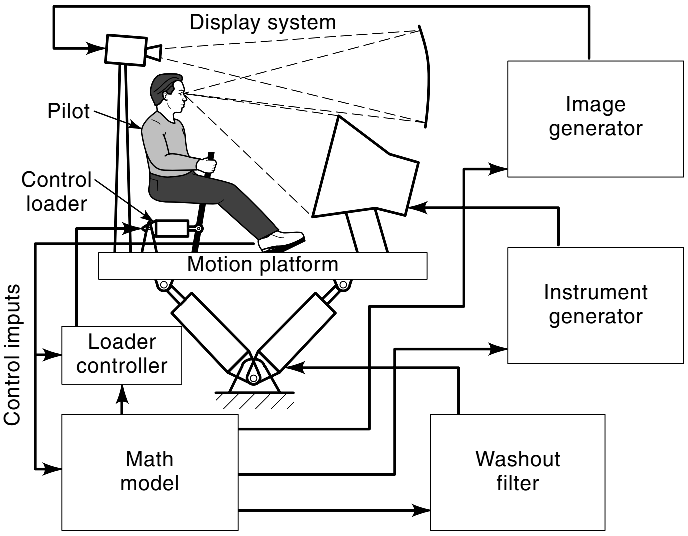

Figure 1. The pilot of a virtual simulator closes several control loops by providing control inputs to the math model in response to cues. The instrument, visual, motion, and tactile cueing systems are illustrated. All are fed state information by the math model.

Figure 1 illustrates the subsystems that create the various cues. Subsequent sections discuss each subsystem.

The engineering premise of virtual simulation is the principle of *physical equivalence*—that is, that identical physical stimuli induce identical sensations and elicit identical responses. Human sensory organs are subject to the laws of physics like any other sensors. Physical replication of stimuli will ensure replication of cues. This is the basis of the present article. The physical nature and accuracy of the replicated stimuli is addressed in objective terms, as might be measured by laboratory instruments. For instance, terms such as "resolution," "adaptation," are used in the optical sense as they would apply to a camera.

Even when the stimuli are perfect, the physical approach is open to challenge on psychological grounds, because the pilot knows that the flight is not real. Actually, the physical stimuli produced by virtual simulators are imperfect, and one is faced with assessing the response of a human subject to incorrect and even contradictory cues. It is impossible to describe virtual simulation without alluding to physiological and cognitive processes. However, our quantitative discussion will be in purely physical terms.

MATHEMATICAL MODEL

An air or space vehicle is a mechanical system. A virtual simulator constructs the state history of this system electronically. The computation must be carried out in real time. In the context of virtual simulation, this means that the computation must keep up with the time history being computed.

At one time, only analog computers were capable of real-time computations. Analog computers are largely limited to linear manipulations, which made it necessary to linearize the dynamic equations. The use of linearized equations lingered even with the advent of digital computers, initially because of their limited capacity and later because of habit. At the present writing, even modest computers are capable of integrating the equations of motion of many aerospace vehicles in real time. It is easier to program the full equations than to linearize them.

The flavor of a typical mathematical model in a virtual simulator may best be conveyed by an overview of the equations governing a rigid vehicle. A rigid body is a six-degree-of-freedom system. The variables of state are

$\vec{x}_e$	Position of CG in earth cartesian system	3 components
$\vec{v}_e$	Velocity of CG in earth cartesian system	3 components
q	Orientation expressed as a unit quaternion	4 components
$\vec{\omega}_b$	Angular velocity in body coordinate system	3 components

These variables are subject to the following equations of motion:

$$\dot{\vec{x}}_e = \vec{v}_e$$

$$m\dot{\vec{x}}_e = \vec{F}$$

$$\dot{q} = \tfrac{1}{2}q\vec{\omega}_b$$

$$J\dot{\vec{\omega}}_b + \vec{\omega}_b \times (J\vec{\omega}_b) = \vec{M}$$

where m is the mass of the vehicle, J is the moment of inertia (a 3×3 matrix), $\vec{F}$ and $\vec{M}$ are the force and the moment applied to the vehicle.

Orientation can be expressed by specifying the heading, pitch attitude, and bank. These three angles, a variation on the ones introduced by Euler to study the spinning top, are called *Euler angles.* This is the preferred formalism for human consumption. However, Euler angles are unsuitable for virtual simulation because they develop singularities at (and lose accuracy near) the orientations of facing straight up or down.

The preferred way of expressing orientations internally in a computer is as unit quaternions. Quaternions are four components entities, which may be viewed as the sum of a number and a vector. Quaternions obey the normal algebraic rules of addition and multiplication with the product of two vectors being given by

$$\vec{U}\vec{V} = \vec{U} \times \vec{V} - \vec{U} \cdot \vec{V}$$

Under these rules, quaternions form a *ring.* All nonzero quaternions are invertible.

A well-known theorem due to Euler states that any two orientations can be bridged by a single rotation. Let the rotation from the reference orientation to the current orientation be characterized by the axis unit vector $\hat{e}$ and the angle α. Then the current orientation may be represented by the unit quaternion

$$q = \cos\tfrac{1}{2}\alpha + \hat{e}\sin\tfrac{1}{2}\alpha$$

This representation has no singularities and maintains uniform accuracy over the entire (curved and compact) three-dimensional space of orientations. However, the constraint $|q| = 1$ must be enforced against truncation errors. Actually, quaternions represent the group SU2 rather than the rotation group SO3, and they cover the space of orientations twice. This detail is of no consequence in simulating a rigid body.

The equations of motion, above, must be integrated numerically. Using advanced–retarded Euler integration with a time step Δt, this is acomplished by the procedure

```
void step(void)
{
Airloads();
t += dt;
Ve += Ae*dt;
Xe += Ve*dt;
Omegb += Jin*(Mb − Omegb^(J*Omegb)*dt;
q += (q*Omegb)*(0.5*dt); q = q/abs(q);
};
```

This is actual C++ code, making use of the user-defined types (classes) of vector, matrix, and quaternion. The global variables of state are declared as

```
void step(void)
{
double t;
vector Xe, Ve, Ae, Omegab;
matrix J;
quaternion q;
};
```

The symbol ^ denotes the vector product. Arithmetic operations are overloaded for the user-defined types. Thus * denotes the product of numbers; of a number by a vector, a matrix or a quaternion; of a matrix by a vector; of two matrices; or of two quaternions. The compiler determines the correct operation based on context. For the product `q*Omegb` (a quaternion by a vector), the compiler converts the vector to a quaternion and employs quaternion multiplication. The overloaded operations of addition and multiplication of vectors, matrices, and quaternions are defined in appropriate header files (1).

The procedure `Airloads()` computes the earth acceleration `Ae` and the body moment `Mb`. Aerodynamic computations are usually based on tables of coefficients and on the local flow field. Often, steady-state aerodynamics for the instantaneous state is used even in transient conditions (adiabatic assumption). Computational fluid dynamics (CFD) is, at this writing, incapable of real-time performance.

Methods of integration more accurate than Euler's are often employed. The powerful Runge–Kutta methods are not suitable when control inputs are sampled only once per step. However, the Adams–Bashforth methods that infer trends from previous steps have been used to advantage.

In many cases, describing the vehicle as a rigid body is not adequate. Examples include helicopters, where flapping and flexing of rotor blades is important, and large aircraft and space structures, where structural modes interact with the control dynamics. In these cases, additional state variables and additional equations of motion are brought into play. The engine and other systems require modeling, too.

TIMING ISSUES

The computation cycle including the sampling of control inputs, the supporting calculation of forces and moments, the integration over a time interval Δt, and the output to the instrument, visual, motion, and tactile cueing systems is called a *simulation frame.* All the computations for the frame must be accomplished within the time period Δt.

Timing may be accomplished by clock-generated interrupts at an interval of Δt. The interrupt starts the frame. Once the frame is complete, computation is suspended until the next interrupt. This method ensures precise timing but, inevitably, wastes some capacity. Another approach is to run the frames continuously and adjust Δt to agree with real time. This ensures the smallest possible Δt while maintaining real time on the average, although individual frames may vary slightly.

The time step used in integrating dynamic equations must not be excessive, in the interest of accuracy. Models of flexible and articulated vehicles place additional burden on the host computer, due not only to the additional degrees of freedom but, more significantly, to the higher frequencies that come into play. The rule of thumb is that the frame rate must be at least ten times the typical frequency of the system being modeled. Frame rates for modeling rigid vehicles are typically between 30 and 60 frames per second (fps). However, for helicopter rotors, frame rates as high as 120 fps are common.

The frame rates of different subsystems of a simulator need not be the same. Even when the dynamic computation requires 120 fps, the visual display may be adequate at 60 fps or even 30 fps, while the motion system and control loader may run significantly higher frame rates, sometimes as high as 5000 fps. The frame rates of subsystems must be commensurate when precise interrupt-driven synchronization is implemented.

Another timing issue involves the interval between control input and observable feedback. The key concepts here are (2,3):

- Latency—the excess delay of simulator response over flight vehicle response
- Transport delay—the delay between control input and simulator response, including computation time but excluding any modeled delay

The transport delay is easier to determine, because it does not require access to the flight vehicle. If the math model is perfect and reproduces the delay inherent in the vehicle exactly, then the transport delay is equal to the latency.

The principle of physical equivalence requires zero latency. It is impossible to have the transport delay at zero, because computations do take time. Some compensation is achieved by not modeling the propagation time of control signals in control rods, wires, and hydraulic lines (at the speed of sound in the particular medium). Still, control responses in virtual simulators are typically delayed.

The pilot expects feedback to control inputs. If this feedback is delayed, the pilot may be induced to increase the input. A delay in any cue will tend to exaggerate the control inputs. In the context of harmonic inputs and disturbances, the delay is translated into a phase lag and it limits the frequency of disturbances that can be controlled.

The FAA accepts a latency of 150 ms for airplane simulators (2) and 100 ms for helicopter simulators (3) for level D certification. Practical experience indicates that simulators subject to this amount of delay are effective. The helicopter value, 100 ms, is representative of the state of the art at this writing. Current simulators of high performance military aircraft also keep the transport delay to less than 100 ms.

Apart from the amount of the delay, there is the issue of the relative delay of different cues. The relative timing of visual, aural, and motion cues is important. Cues received out of order may cause *simulator sickness*—a condition where an experienced pilot becomes nauseated in the simulator.

COCKPIT DISPLAYS

Flight and engine instruments are the cueing devices that are easiest to implement in a virtual simulator. The Link trainers of WW II fame used analog computers to drive needles in electrically actuated replicas of airspeed indicators, altimeters, tachometers, and other airplane instruments. The devices were used to teach control of an airplane by sole reference to instruments, which made visual displays unnecessary. The Link devices had rudimentary motion capability of questionable fidelity. The task was to train a pilot deprived of visual cues to disregard motion cues and react to instrument readings only. A number of postwar fixed-base devices whose general architecture was the same as that of the Link device accomplished the same end. They were useful in teaching instrument flight and maintaining instrument proficiency and were accepted by the FAA for training and currency credits.

With the advent of microprocessor technology, even low end simulators became digital. Computer graphics made it possible to use graphical images of instruments in place of hard replicas. The first FAA-accepted device to exploit this capability was the Minisimulator IIC, which came on the market in 1981. The IIC used most of its computational throughput to create the crude two-dimensional graphical representation of the instruments. But graphics techniques soon improved, and graphically displayed cockpit instruments became commonplace in actual cockpits as well as in simulators.

In addition to instruments, many modern cockpits include other displays. Some, like moving maps and horizontal situation displays, are two-dimensional. Others, such as low-light-level TV (LLTV) and forward-looking infrared (FLIR), offer a view of the three-dimensional outside scene. The three-dimensional graphic displays are computed by the same methods as visual displays discussed in the next section.

IMAGE GENERATION

Creating a visual display of the outside scene is by far the most computationally demanding task in a virtual simulator. Early image generators (IG) used analog methods. A television camera would "fly" under computer control over a miniature scene or an aerial photograph. Early digital image generators offered night scenes with only discrete points of light visible. The technology soon advanced to dusk and eventually to daylight scenes.

Data about the three-dimensional environment in which the flight takes place is kept in a database. Terrain and other objects are described as "wireframes" delimited by polygons. Each polygon is endowed with color and/or texture. There have been efforts to create an open database format; at this writing, the formats in use are mostly proprietary.

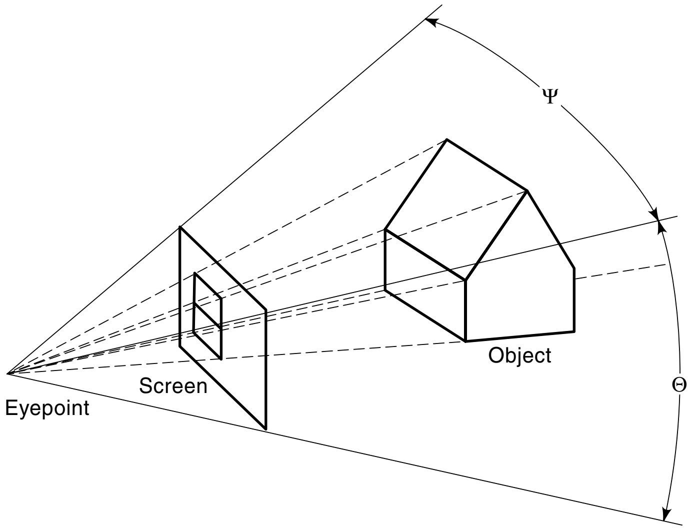

Figure 2. The three-dimensional scene is transformed into a two-dimensional graphic on the image plane by projecting along rays that meet at the eyepoint.

The screen image is a projection of the three-dimensional scene on an imaginary screen by rays converging at the intended viewer's eye (Fig. 2). Different shapes of the two-dimensional display are in use. However, for simplicity, this discussion addresses a rectangular screen which is placed to subtend a pre-selected field of view (angle Ψ by angle Θ in Fig. 2). When the image is presented in the simulator (next section), it should cover an equal portion of the pilot's field of view.

The methods employed by image generators for flight simulation are similar to the ones used in other computer graphic applications that produce perspective views of three-dimensional objects. The specific tools used by manufacturers of IGs are usually proprietary. The overall approach is best illustrated by the OpenGL language (a public domain offshoot from the proprietary IrisGL) (4).

OpenGL supports a transformation formalism based on 4 × 4 matrices. These represent not only the Euclidean group (translations and rotations) but also affine and projective transformations (5). This formalism supports the projection shown in Fig. 2. It can also create an image so that it appears correct when projected from one point and viewed from another. This is pertinent with front projections, since the projector and the pilot's eye cannot be collocated. The transformation between the projector image and the viewed image is known as "distortion correction." A more complex instance of distortion correction arises with spherical screen. High-end image generators perform that transformation, too.

The projection in Fig. 2 represents "one channel" of an image generator. The image generator may offer several channels. A wide field of view may be created as a mosaic of several adjacent or partly overlapping channels. Still, the field of view in a simulator is usually restrictive in comparison with the aircraft.

A typical channel might drive a raster display of 1280 pixels by 1024 pixels and cover a field of view of 40° × 30°. This choice makes each pixel subtend 1.9′ (1.9 minutes of arc) and effectively limits the simulator pilot to 20/40 vision. Physical equivalence would dictate that 1′ be resolvable, corresponding to the 20/20 vision required of airmen. However, 2′ or even 3′ resolution is representative of current simulator practice.

Many arguments can be raised to rationalize the contradiction between accepting a 20/40 or 20/60 simulator while insisting on 20/20 vision for the pilot—for example, that the performance of most individual tasks does not really require 20/20 vision and that most of the collective training in simulators is for night and adverse weather conditions. In reality, this policy is driven by supply and demand. A 20/20 simulator would be exorbitantly expensive, whereas humans with 20/20 vision are plentiful.

The display for our typical channel consists of 1,310,720 pixels. The image generator must specify the color of each pixel. At 32 bits per pixel, the "local buffer" comes to 5.24. A double buffer is required for smooth operation: the image generator redraws the picture in a hidden buffer, leaving the one being displayed undisturbed. Once the updated picture is complete, the buffer pointers are switched. Thus a channel requires 10.5M of memory.

A depth buffer is also required. It is also called "z buffer," because, by convention of computer graphics, the coordinate system is oriented so as to make the depth (the distance from the viewer) the z coordinate. The z buffer is a scratch pad that the image generator keeps for itself. The depth of the surface that generated the pixel stored in the local buffer is kept in the z buffer. For each pixel, the image generator goes over the database and, for each polygon, determines whether that polygon is intersected by the ray corresponding to the given pixel. If so, its depth z is computed and compared with the value in the z buffer. If the new object is closer, the pixel is rewritten to represent it; otherwise it is not. The buffers are initialized to a background color in the local buffer and "infinity" in the z buffer. The z buffer occupies another 5.24M. This amount of memory has become commonplace. But the task of reworking it at the required rate is still a challenge.

The picture, which appears to be moving continuously, is computed as discreet images. Each image is traced over the screen in a manner similar to other computer displays. Two rates govern:

- The refresh rate, at which the screen is retraced
- The update rate, at which the content of the picture is updated

A refresh rate of 60 Hz or higher eliminates "flicker." The refresh rate also sets a practical bound on the update rate. Even when dynamic computations are carried out more often, the additional information cannot be displayed visually. However, the update rate can be lower than the refresh rate.

The smoothest motion is obtained when update and refresh are synchronized—that is, when the refresh rate is divisible by the update rate. For a refresh rate of 60 Hz, this consideration would allow an update rate of 60 Hz, 30 Hz, 20 Hz, 15 Hz, 12 Hz, ...

The update rate that is required in a simulator varies with the task being simulated. Most demanding are the tasks that involve rapid change in the scene. This occurs during rapid angular motion of the vehicle. In the case of head- or helmet-mounted displays (next section), rapid change in the scene can be caused by brisk head movements. An update rate of 60 fps is adequate for most purposes. Lower rates are acceptable in many cases. Sensitivity to update rate varies with the individual subject.

Image generators are required to perform several additional tasks:

- *Moving Models and Articulated Parts.* Other vehicles must be displayed at changing locations. In some cases, articulated parts, such as the turret on a tank, must be seen moving.
- *Terrain Elevation and Slope.* It is the image generator that has direct access to the model of the terrain. The host requires the terrain elevation and possibly slope for use in the ownship ground contact model. It is up to the image generator to supply these.
- *Color Variations.* The same scene may need to be rendered in different colors to represent day, night, or sensor images.
- *Text and Symbology.* At times, it is desired to have the image generator create the text and symbols contained in a heads-up display (HUD) and superimpose them on the scene.

All of the above have negligible impact on image generator performance. The cost of rendering the polygons making up the moving models and parts is the same whether they are moving or not. The added burden for all tasks, above, is on the communications of the IG with other computers.

Other functions, such as accurate simulation of light sources with correct rendering of shadows and highlights, are demanding of the IG. Correct simulation of limitations to visibility by mist or smoke is likewise expensive. On the other hand, a crude representation of the same effects is helpful in that it eliminates the labor of rendering the obscured objects.

DISPLAY SYSTEM

The two-dimensional projection of Fig. 2 must be presented to the simulator pilot within the original viewing angles Ψ and Θ. This may be accomplished by a small image nearby or a larger image further away (Fig. 3). It may be a "real image" projected on a screen or traced on a CRT or a "virtual image" created by optics. A real image is limited by practical considerations to be within a few meters from the eyepoint. A virtual image can be as far away as desired and even infinitely far. With the pilot's eye at the eyepoint, all the images in Fig. 3 create the same impression on the retina, with the same resolution. But there are significant differences:

- *Accommodation.* The pilot's eye must accommodate optically to the distance at which the image is located rather than the real-world distance of the objects it contains. Should the pilot need corrective lenses to aid in accommodation, these would not necessarily be the same in the simulator as in flight.
- *Parallax.* Even when seated, the pilot's upper body and head is free to move to some extent. As the eye moves, nearby objects (e.g., the cab environment) change their apparent position relative to objects that are further away. With the simulator display this would be governed by the distance of the image rather than the distance to the objects it represents. Objects in the image will not move relative to each other. Should the pilot's eye deviate from the nominal eyepoint, the perspective would become distorted. During forward flight this would create the impression of a spurious sideways component of motion.
- *Stereopsis.* When the pilot's two eyes observe the same image from slightly different vantage points, the two retinal impressions differ. This difference is the raw material for stereopsis, which determines apparent distance. The distance so determined is that of the image rather than of the objects it represents. The stereopsis cue might conflict with other cues—for example, perspective cues and cues based on the size of familiar objects.

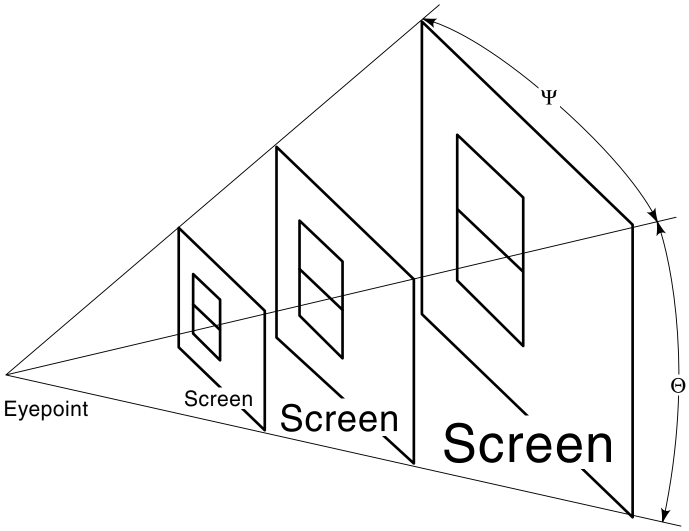

Figure 3. Image planes at varying distances from the viewer create the same impression on the retina, with the same resolution. But accommodation, parallax, and stereopsis effects differ and betray a close by image for what it is—a small, flat picture.

These effects are most pronounced with a small, nearby display, such as a monitor screen. They flag the image as a small, flat picture. A human being can transcend this detail when appreciating art. To some extent, one can transcend it during training of specific tasks. Screen displays as close as one meter have been used successfully and accepted well by experienced pilots. However, to attempt physical equivalence, one must do better. This is where the display system comes in.

A screen projection is a significant improvement over a monitor screen. The image may be projected either from the front of the screen or, with a suitable screen, from the rear. *Back projection* has the advantage that the projector is out of the way of the pilot and cab structure. It is possible to place the projector so as to avoid distortion and the need for distortion correction.

A larger image placed, typically, three meters away is easier to perceive as real. The accommodation is only 0.3 diopter from infinity. Parallax with nearby objects, such as the cockpit structure and instruments, is approximately correct.

Infinity optics is a more effective solution. The image is optically placed infinitely far away. Accommodation is exactly correct for distant objects as is parallax with the cab environment.

To avoid color fringes, infinity optics must employ mirrors rather than lenses. A collimator, illustrated in Fig. 4, is a common example. The monitor is set at 90° to the pilot's line of sight. A "beam splitter" semireflective glass plate, set at 45°, reflects the monitor screen into the concave spherical mirror. The pilot views the mirror through the beam splitter. The monitor face is at the mirror's focal point (half radius as measured along the broken optical path). Light originating from a point on the monitor comes out of the mirror as a parallel pencil of rays, putting the image out at infinity.

A collimator typically covers the field of view of one channel. Three channels may be combined by a battery of three collimators set at an angle to each other. Such batteries are designed with "overfill." This means that the pictures in adjacent monitors overlap. When the pilot's head moves, parts of the scenery that were near the edge of one collimator are now seen in the other. This way, the three collimators offer a seamless combined view.

The collimated image at infinity can be seen only when the viewer's eye is within the fairly narrow collimated beam. Collimators act as funnels with an opening to the distant scene. Eyepoint movement does not distort the scene, but excessive movement blocks it. Two pilots cannot share a collimator. They must be given two separate collimators even when the same IG channel drives both with an identical image. Supplying more than one crewmember with a wide field of view is impractical because of mechanical interference of the systems of collimators.

Collimators cannot match the field of view offered by a spherical dome that encloses the pilot and makes a borderless projection screen. But sharing of a dome or screen projection by two crew members is problematic. The basic image at infinity is the same, but the distortion correction is different for the two eyepoints.

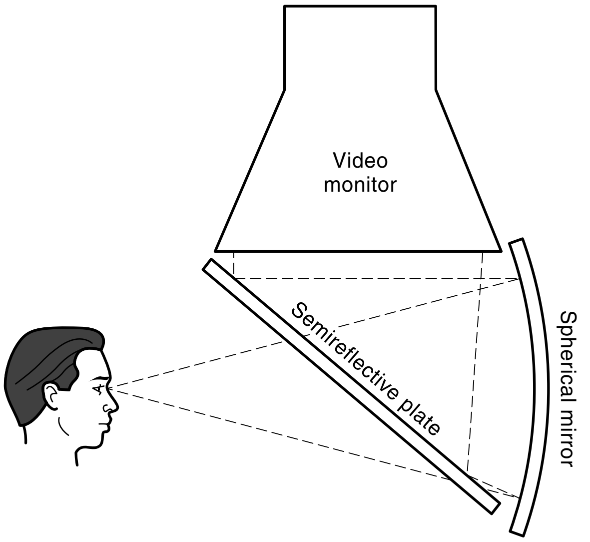

Figure 4. A collimator serving as infinity optics. The monitor faces down. The screen is reflected into a concave spherical mirror by a diagonal semi-reflective glass plate. The pilot views the mirror through the plate. The mirror creates an image located at infinity.

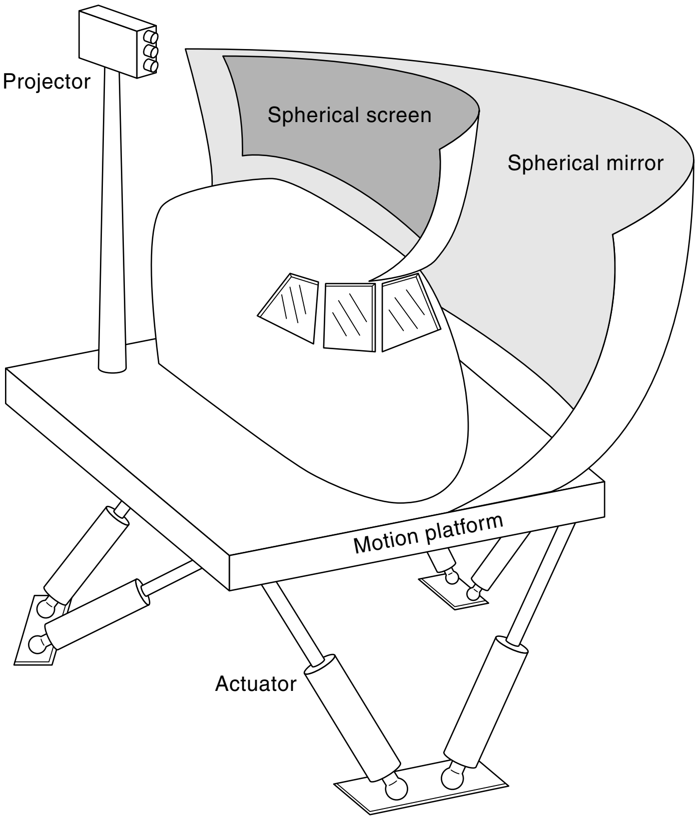

Figure 5. A six-post motion platform is capable of six DOF motion. The platform carries a simulator cab and a display system with wide-angle infinity optics. The display system employs back projection on a spherical screen which the crew views reflected in a large spherical mirror.

Figure 5 shows an elegant solution: an infinity optics system that can serve several crewmembers and provide them with a correct, wide-angle outside view regardless of their position in the cockpit. The picture is back-projected by a number of projectors (only one is shown) onto a spherical screen. The simulator crew views this display through a large concave spherical mirror. The screen and mirror are concentric with their radii matched to put the screen at the focal surface of the mirror as viewed from the cab. The mirror creates a virtual image located out at infinity that can be seen from anywhere in the cab.

Neither the projected image nor the one viewed through infinity optics offers correct stereopsis, parallax, or accommodation for objects that are not far away. This is significant for operations where nearby objects play a role, including aerial refueling, spacecraft docking, and maneuvering helicopters near terrain and objects.

Stereopsis can be achieved by offering separate images for the two eyes. When this is done, the stereo cue is expected to overpower the accommodation cue and the parallax cue with which it is not consistent.

Three-dimensional images that are inherently correct in stereopsis, accommodation, and parallax for any viewer and for multiple viewers at the same time can be produced by holography. But holography requires creation of an interference pattern with resolution of the order of the wavelength of visible light (in the order of 10^{-8} m). This capability is not yet available in real time.

Separate images for the two eyes (or for that matter, for two crew members) can be offered with projection systems and infinity optics systems by use of polarized light or of electronically timed shutters. In the former case, two separate images are projected on the screen using mutually orthogonal polarization. The pilot views the display through polarizing lenses, so that each eye sees only one image. In the latter case, the two images alternate. The pilot views the display through electronically timed liquid crystal shutters. These block each eye when the image intended for the other is projected.

Head (or helmet)-mounted displays (HMD) offer separate collimator-like display systems for the two eyes. The HMD requires head tracking to determine the instantaneous orientation of the eyepoint. Head movement can sweep a narrow field of view over a much wider field of regard. These systems typically induce the pilot to substitute head movement for eye movement, and the natural ability to notice moving objects in one's peripheral vision cannot be exercised.)

The quality of HMD depends on the precision of head tracking and its latency. The display requires a fast update rate to keep up with fast image changes due to abrupt head movement. HMDs typically require individual fitting. The size and weight of an HMD is a burden on the civilian pilots. Even military pilots, used to flying with a helmet, often object. Besides, the HMD precludes the use of operational helmets and viewing devices in the simulator.

The eyepoints used for the HMD are generic. They represent the eye positions of a typical pilot. Static adjustment to the pilot's seat position, torso height, and eye separation is feasible. Dynamic adjustment to body and head movement is not in the current systems.

For use with an HMD, the database models the inside of the cab as a black silhouette. The HMD reflects its images on beam-splitters that allow the pilot to see through into the cab. Even so, there is a potential problem when two crew members sit side by side. The silhouette of the other crew member's head cannot be predicted perfectly and will not register accurately. Bright outside scenery may "show through" the edges of the other crew member's helmet.

Brightness is an issue for all simulator displays. One must assess the brightness available at the source and how much of it reaches the observer's eye through the display system optics. These estimates are too involved to be presented here. The bottom line is that there is no difficulty in creating what an observer will accept as a daylight scene. The brightness of this scene is far below actual daylight. Pilots do not use their sunglasses in simulators. Simulator cabs are darkened during operation unlike aircraft cockpits in daytime. By the same token, problems of observing certain dimly lit displays in sunlight do not arise in the simulator.

It was not possible to describe in this section all the types of display systems in current use. Some of the ones not covered are calligraphic displays, multi-resolution displays, and area of interest displays.

MOTION CUES

Motion cues, by definition, are those cues that result from the pilot being moved bodily. Awareness of motion through sight or sound is excluded. It is a basic law of nature that, without reference to external objects, uniform rectilinear motion is undetectable. It is also a basic law of nature that, without external reference, the effect of acceleration is indistinguishable from that of a gravitational field. What is locally measurable is *specific force,* which is an effective acceleration of gravity given by

$$\vec{s} = \vec{g} - \vec{a}$$

where $\vec{g}$ is the local acceleration of gravity and $\vec{a}$ is the acceleration of the cab relative to an inertial system. Rotation relative to an inertial frame is also measurable.

It is these parameters, namely, the three components of specific force and the three components of angular velocity, that serve as motion cues for the human body, as for any other physical system. The inner ear contains organs (otholiths and semicircular canals) specifically adapted to sense these parameters. The motion parameters are felt also by other parts of the body—for example, the sinking sensation in the pit of the stomach when an elevator starts its descent. So long as the six motion parameters are reproduced correctly, there is no need to investigate the mechanism of human perception. Any and all mechanisms respond as they do in flight.

It takes a six-degree-of-freedom motion system to create the six motion cues. With the simulator cab on a motion platform, the pilot can sense rotational rates around the three body axes (yaw, pitch, and roll) and linear acceleration forward, sideways, and up (surge, sway, and heave). The six parameters vary from one point in the moving cab to another. However, with a rigid cab representing a rigid vehicle, if the parameters are correct at one point, they are correct at every point.

When the replication of the motion parameters is only approximately correct, the errors vary from point to point in the simulator cab. It is then necessary to select a *sensing point* where the errors are minimized. The choice of a sensing point is influenced by the theory of perception. For example, if it is the inner ear which processes the motion cues, then the sensing point should coincide with the pilot's head.

The fact that uniform motion is intrinsically undetectable allows a pilot to have the same sensations in a stationary simulator as in a fast-moving airplane. However, acceleration and rotation are sensed. It is impossible to replicate the acceleration of the flight vehicle exactly while keeping the motion platform in the confines of a room. For instance, during the takeoff run, an airplane accelerates from rest to flying speed. In the process, it might roll over a few thousand feet of runway. Should the motion platform be subject to a surge acceleration equal to the airplane's, it, too, would translate a few thousand feet and out of the confines of the building that houses the simulator.

The above discussion demonstrates that a confined motion platform, of necessity, violates the principle of physical equivalence under some circumstances. One attempts to replicate the motion cues approximately, and, to the extent possible, deviate from the true motion parameters to a degree that is undetectable by a human subject.

In the case of the takeoff roll, the specific force, in body coordinates, is inclined to the rear and is slightly larger than

1 g. The motion platform, confined to a small space, cannot replicate this condition. The platform can be tilted to a nose-up attitude so that the direction of the specific force is correct. The magnitude remains 1 g. However, the small difference may not be obvious to the pilot. There remains the problem of achieving the tilt at the onset of acceleration. This must be done slowly, at a rate below the pilot's threshold of detection.

The translation of the vehicle motion as computed by the math model to a motion command for the motion platform is accomplished by a *washout filter.* The functions of the washout filter are to:

1. Limit commanded linear and angular motions to platform capability
2. Slow the platform near its limits to avoid banging into the stops
3. Stealthily return the platform to mid-range
4. Tilt the platform to simulate sustained surge and/or sway acceleration

Items 3 and 4 should be accomplished at rates below the pilot's detection threshold. The tilt due to item 4 should be combined with the instantaneous orientation so as to ensure the correct direction of specific force. The equations for accomplishing this are delicate. Many practical simulators approximate this procedure by merely superimposing Euler angles.

The most common configuration of a motion base is the "synergistic" or "six-post" arrangement. Motion platforms of this design are used for both training and engineering simulators. As shown in Fig. 5, the platform carries the simulation cab and the visual display system. A high-end "six-poster" might be rated for a load of 2 tonnes or 3 tonnes. It can provide linear accelerations as high as 1 g, angular accelerations as high as $150°/s^2$. However, linear displacements are limited to under 1 m and angular displacements to 15° or 25°. Some unique motion platforms at research facilities can do better. The Vertical Motion Simulator (VMS) at the Ames Research Center of the National Aeronautics and Space Administration (NASA) allows 2.4 m of surge, 12 m of sway, and 18 m of heave.

Motion systems are also characterized in terms of response to harmonic inputs. The recognition of particular motion cues, such as the bumping of the left main tire against the runway, depend on undistorted transmission of fairly high frequencies, up to about 50 Hz. For this reason, the computation systems driving the motion platform must compute at a rate of $\approx$500 fps or higher. Until quite recently, analog systems were used to meet this requirement.

The phase delay is a separate issue, which is pertinent for motions that the pilot manually controls and damps. A human subject cannot do this consciously above $\approx$1 Hz and probably a little higher for subconscious tasks. The phase lag is a direct function of the latency. A 100 ms delay translates into 90° of phase lag at 2.5 Hz. Typically, the phase lag reaches 90° at 1.5 Hz or less.

The frequency response depends on the mechanical system driving the motion platform as well as the computer system and the washout filter. Most high-quality motion systems are driven hydraulically. However, electric systems have advanced recently and now occupy the low end of the price range.

The motion amplitudes of a six-post platform is sufficient for simulating a transport aircraft that maneuvers gently. (However, the very low frequency heave cues in the landing flare may be truncated.) The VMS has been used extensively in the study of helicopters. Neither system is capable of the sustained high specific force ("high g") that fighter aircraft develop during steep turns and other vigorous maneuvers. These can be developed by different designs of motion platforms that act as a centrifuge. However, the unwarranted high rate of rotation in a centrifuge presents a problem.

Another condition that a confined simulator cannot sustain is that of 0 g, or weightlessness. It can be sustained for a minute or two in an airplane flying a parabolic trajectory. This was used by NASA to expose the astronauts to weightlessness in advance of space flight.

The ability of an aircraft to induce a wide variety of specific force conditions suggests the *in-flight simulator*—the use of an aircraft as a flight simulator. One aircraft can simulate another. The astronauts learned to control the Space Shuttle in a modified Gulfstream G2 that simulated it. However, the subject of in-flight simulation is outside our scope here.

Most military simulators of fighter aircraft are fixed base. Cues of high specific force, typical of fighter aircraft, are transmitted to the pilot through the pressure suit that fighter pilots wear. "High g" has the effect of driving the blood into the legs and lower body and away from the brain. The pressure suit counters this effect by increasing the pressure over the legs and lower body in response to the level of specific force. In a simulator, even though the specific force remains at 1 g, the suit inflates in response to the computed specific force and provides the pilot with a g cue.

Use of the pressure suit is a blatant deviation from the principle of physical equivalence. Rather, it is an application of the psychological phenomenon of *association.* The pilots have become accustomed to associating the suit pressure with high g effects. When the suit inflates, the human subject, in the manner of Pavlov's dogs, may imagine that high specific force prevails.

There are other pseudo-motion devices in use. One is the pressure cushion that inflates when increased g is computed. The cushion is supposed to simulate the increased pressure on the pilot's buttocks. Increased pressure may be experienced when the pilot's seat belt is secure. Squeezing the tissues between the seat and the belt is not physically equivalent to the effect of increased g. But the pilot does get a cue.

The subject of motion in flight simulation is controversial. The FAA insists on motion. Devices without motion are classified as "training devices" rather than "simulators" and allocated reduced credits. The utility of fixed base simulation is well established in the military and elsewhere.

Cases where motion degrades a simulator and induces motion sickness have been observed. The probable explanation is that "bad motion is worse than no motion." Motion can be "bad" because it is a poor emulation of the specific force and angular rates experienced in flight; or because of excessive latency; or because it is poorly synchronized with the visual cue; or because it betrays the mechanics of the motion base.

How good should motion be? Some idea of the answer may be conveyed by Ref. 6. This experiment used the sway-and-roll motion of the VMS in a sequence of side-step maneuvers. The data demonstrated the importance of the motion cue. However, scaled-down motion produced objective performance equal to the full-scale motion and got better subjective evaluation from the pilots. Not even the VMS was capable of altogether "good" full-scale motion. The motion system of the VMS has been upgraded in the wake of the Ref. 6 results.

When consistent motion and visual cues are available, the motion cues should be sensed by the pilot earlier. An acceleration step of magnitude a results in a displacement $\frac{1}{2}at^2$. This displacement is not sensed visually until it has grown to the visual detection threshold Δx, which takes a time delay

$$\Delta t = \sqrt{\frac{2\Delta x}{a}}$$

So long as a is above the acceleration detection threshold, the specific force due to the acceleration is felt immediately.

The importance of the motion cue varies with the task being trained. In my judgment, it is most significant with tasks performed subconsciously. The landing flare of a fixed-wing aircraft and the hovering of a helicopter (with no stability augmentation) may rely significantly on motion cues.

CONTROL LOADING

In the early days of aviation (and to this day in light aircraft), pilot controls were coupled mechanically to aerodynamic control surfaces. Pilots relied on the control feel, dominated by aerodynamic forces, as a major cue. The function of the control loader is to reproduce this cue in a flight simulator.

In the meantime, aircraft have evolved. Hydraulically actuated controls have become the norm. Electronic controls are the trend of the future. These irreversible control systems do not feed aerodynamic forces back to the pilot. Artificial feel systems (usually springs) are used to provide the pilot with a semblance of the expected feel. Increased reliance on instrument readings makes up for the deficiency.

Control loaders are fairly expensive. A high-quality loader may cost more than a light airplane. This creates a paradoxical situation: a control loader can be economically justified only in those cases in which the most important cues that it can provide are suppressed. A very sophisticated piece of equipment simulates a generic system of two masses, with springs, dampers, and linkage. This is traditionally approximated by a near linear model. Nevertheless, control loaders are important in special situations—for instance, hydraulic failure, giving rise to significant control forces.

The techniques of control loading are similar to the ones employed in motion system. The high-end control loaders are hydraulic, with electric systems starting to catch up. Through the 1980s, control loaders were controlled by analog computers. In the 1990s, digital controllers caught up, some of them using frame rates as high as 5000 fps.

THE VIRTUAL COCKPIT

Flight simulation may be viewed as a precursor and special instance of the emerging technology of virtual reality (VR). A flight simulator creates a near physically equivalent virtual environment for flight crews enclosed in the cab. VR tends to avoid physical props. It would dispense with the physical cab and replace it with a virtual cockpit. Historically, VR is younger than flight simulation, and the two communities are largely disjoint.

VR provides visual cues through an HMD. The image generator, rather than blocking the inside of the cab, would include it. The HMD would provide the pilot with a stereoscopic image of the inside as well as the outside of the cab. Tactile cues are produced by devices, ranging from gloves to "exoskeletons," attached to the subject's person. These are tracked, and they are controlled to apply appropriate forces to the human body.

The obvious benefit of this plan is that it would make the simulator generic. Reconfiguration to any type of vehicle becomes selectable by software. But there are many technical problems to be resolved. All the drawbacks of HMDs mentioned previously apply, and their effect is magnified in relation to the closeby cockpit scene. The gloves and other tactile devices are yet to be proven. Progress will probably start with the instrument displays and move to switches and secondary controls. The physical major flight controls and physical seat will be last to go.

VR has been employed in aerospace for prototyping the interiors of vehicles ranging from airliners to the space station. VR, together with visual and motion devices borrowed from aerospace simulation, are making a splash in the entertainment industry. Lay subjects enjoy exciting sensations of presence and motion. But experienced pilots, conditioned to true visual and motion cues, are more critical.

NETWORKING OF SIMULATORS

Long-haul networking came into its own in the 1990s. Air combat simulators with dual cockpits engaging one another have been in existence since the 1960s. By the 1980s, several simulation facilities had connected their simulators by a local area network. The concept was taken a step further by the Defense Advanced Research Projects Agency (DARPA). In the SIMNET project (7), large-scale networking, including remotely located facilities, was carried out successfully.

The SIMNET project used low-fidelity simulators with crude visual displays. Active controls and instruments were limited to the ones normally used or monitored during combat. Everything else was eliminated or represented by static props and pictures. The purpose was to recreate the feel, the pressures, and the confusion of a battlefield. In a test conducted in 1989, about 400 players participated, including tank crews and helicopter crews at separate army installations.

SIMNET achieved its networking in two stages. Local networking tied simulators within one facility together by use of Ethernet. The long-haul link between different facilities used commercial 56 kbaud lines. The local and long-haul protocols were different.

Like the local networking that preceded it, SIMNET addressed a set of matching simulators specifically designed to interact. By 1989, there were also isolated demonstrations of long-haul communications between existing high-fidelity simulators that were separately and independently designed and owned. In 1979, an F-15 simulator located at Williams Air Force Base engaged an F-4 simulator at Luke Air Force Base. Both bases are in Arizona, and the distance between them is 80 km. The network link used four telephone lines.

In 1989 a long-haul link between an AH-64 Apache simulator located in Mesa, Arizona and a Bell 222 simulator located in Fort Worth, Texas was demonstrated. The Arizona simulator was in the facility of the McDonnell Douglas Helicopter Company. The Texas device was in the plant of Bell Helicopter Textron. The distance between the two facilities is 1350 km. The link employed a 2400 baud modem over a standard telephone line.

These experiments showed that long-haul networking of dissimilar simulators was practical. But a communications protocol was missing. Rather than reinvent the interface by mutual arrangement between each pair of facilities, an industry standard for interfacing simulators was needed. By conforming to the standard, a simulation facility could ensure compatibility with every other facility that conformed.

An open industry standard for networking of simulators was first addressed at a conference held in Orlando, Florida, in August 1989 (8). The conference adopted the local SIMNET protocol as the starting point for the new standard. The term coined for the new protocol was distributed interactive simulation (DIS). Work on DIS continued in biannual meetings in Orlando. In 1993, the DIS protocol was formalized as IEEE Standard 1278-1993 (9). Work on upgrades continues.

The number of players involved in SIMNET was large enough to enforce some of the mandatory rules of large scale networking: The participating simulators must be independent. Each must be able to join the game or withdraw without interfering with the operation of the others. The failure of any single simulator must not disrupt the game.

But the SIMNET protocol also involved design decisions tailored to the low processing power of the SIMNET devices. Some of these design details were not desirable in general. The lessons of the long-haul SIMNET protocol were lost and had to be relearned.

The technical challenges of long-haul networking are mostly two: bandwidth and transmission delays. These issues exist in local networking, but long distances between networked simulators render both issues more critical.

When a large number of simulators interact, current state information about each vehicle must be broadcast for the benefit of all. Broadcasting all this information at the rate at which it is created—typically 40 to 60 times a second—creates prohibitively large information flows. Methods for reducing the required bandwidth were needed.

One method, introduced in SIMNET, is called *dead reckoning*. This term, borrowed from navigation, refers to the extrapolation of a vehicle's motion based on its previously known state. The SIMNET dead reckoning scheme has each simulator withhold its broadcasts so long as its state information can be reproduced with acceptable accuracy by extrapolation. The originating simulator (the sender) determines whether this is the case by simulating the extrapolation process of the remote simulator (the receiver). For each simulation frame, the result of the extrapolation is compared to the state of the vehicle computed for that frame. No broadcasts are made until the difference exceeds a preselected threshold.

Other methods for relieving the bandwidth bottleneck include (a) bundling of packets at each node and (b) long-haul transmission of changed information only.

The second technical issue is delay. Remote information is outdated information. A delay corresponding to the speed of light is a hard minimum imposed by the laws of nature. It amounts to 3.33 μs/km. Over global distances of several thousand kilometers, the delay is comparable to a simulation frame. The delay in actual communications lines is roughly double the above. With a satellite link, the round trip to geostationary altitude imposes a delay of 200 ms, and the mechanics of the equipment on the satellite increases this to half a second or more. Further delays are caused by processing packets by servers at network nodes.

An aircraft traveling at 400 knots covers 1 m in about 5 ms. A rotorcraft flying at, say, 100 knots takes 20 ms to cover 1 m. Position discrepancies due to communications delays are visible in close formation flying. Hit-or-miss decisions for projectiles are affected.

Delays in communications channels are not predictable and not repeated precisely. A constant delay will make the remotely simulated vehicle appear to lag behind, whereas a variable delay will make it appear to jump around. To compensate for the delay, remote data must be extrapolated to the current time over the delay period Δt.

Initially, there was the misconception that, so long as sender and receiver used the same dead reckoning scheme, the receiver error would never exceed the threshold imposed by the sender. The fallacy of this view was soon exposed (10). The sender withholds its broadcasts until after the threshold has been exceeded. At that time, the sender broadcasts an update. But the update does not reach the receiver until Δt later. All this time, the receiver's error continues to grow.

Even when the update arrives, the receiver is not at liberty to exploit it. Immediate reversion to the more recent data would cause a visible jump in the image. This would make the image jitter and betray that it is the image of a remotely simulated entity. The receiver must implement *smoothing*. Depending on the particular smoothing algorithm, the receiver will maintain the state error longer or even continue to grow it for a while after the update is received.

This way, the receiver's error always exceeds the sender's threshold, and, in long-haul networking, by a very significant margin (11). Dead reckoning, which, for the sender, is a bandwidth saving device, becomes a mandatory accuracy maintenance procedure for the receiver. Needless to say that dead reckoning by the sender increases the delay and so does any bandwidth saving scheme that requires processing at the nodes.

The receiver must extrapolate the state in each packet over the delay that the packet experienced. To make this possible, it is necessary to include a *timestamp* with the variables of state in each data packet. The stamp is the time for which the variables are valid as opposed to the time at which they were computed or transmitted. The receiver subtracts the timestamp from the time at which the variables are to be displayed and extrapolates over the difference. The error in the dead reckoned state depends on the accuracy of the timestamp as well as on the extrapolation algorithm (10).

The DIS protocol specified a timestamp since the 1990 draft. Two versions of a timestamp were recognized: an absolute timestamp produced by a clock synchronized to universal time coordinates (UTC) and a relative timestamp produced by a free running local clock. The relative timestamp can be used to correct for the jumping around effect of variable delay, but not for the lagging behind that the delay itself causes.

To produce an absolute timestamp, clocks at remotely located simulation facilities must be synchronized to within a millisecond or a few milliseconds. This has been made easy by the Global Positioning System (GPS). GPS time, accurate to better than a microsecond, is available as a biproduct of the position calculation. (GPS time differs from UTC time by a small integral number of seconds accumulated in leap years.) It has been shown that the state error due to delay and to clock error are additive (10). With GPS, the clock error can be effectively eliminated.

Networked simulation in the service of the military has mastered the bandwidth issue and has achieved reliable high-volume operation over its own dedicated Defense Simulation Internet (DSI). Synchronization of simulation clocks is still not prevalent. Facing up to the challenge of specifying, and verifying precision and consistency for long-haul simulation, at this writing, is still pending.

Table 1. Communications Requirements

Issue	Normal Requirements	Simulation Requirements
Acknowledgments	Required	Useless
Transmit queue protocol	Deliver packets in order queued	Deliver most recent packet and discard others
Receive queue protocol	Process packets in order received	Process most recent packet and discard others
Receive buffer full	Halt transmission	Impossible
Transmit buffer full	Halt process	Impossible
Checksum	Required	Required
Corrupted packet	Ask for retransmission	Discard
Lost packet	Ask for retransmission	Forget

COMPUTER COMMUNICATIONS

Virtual simulation requires interprocess communications, be it the local or long-haul networking of simulators or the communications between different processes within one simulator (Fig. 1). Table 1 lists the requirements of asynchronous communications in the service of virtual simulation. These requirements are different from the ones prevailing in other fields. They offer an incentive for simulation specific communications protocols.

BIBLIOGRAPHY

1. A. Katz, *Computational Rigid Vehicle Dynamics,* Malabar: Krieger, 1997.
2. Federal Aviation Administration, Airplane simulator qualification, *Advisory Circular,* **120-40B**: 4–5, July 29, 1991.
3. Federal Aviation Administration, Helicopter simulator qualification, *Advisory Circular,* **120-63**: 2–3, October 11, 1994.
4. J. Neider, T. Davis, and M. Woo, *OpenGL Programming Guide,* Reading, MA: Addison-Wesley, 1993.
5. M. E. Mortenson, *Geometric Transformations,* New York: Industrial Press, 1995.
6. J. Schroeder and W. Chung, Effects of roll and lateral flight simulation motion gains on a side step task, in *Proceedings of the 53rd Annual Forum of the American Helicopter Society,* Virginia Beach, VA, June 1997, pp. 1007–1015.
7. A. R. Rope, The SIMNET Network and Protocols, Report No. 7102 by BBN Systems and Technologies prepared for the Defense Advanced Research Projects Agency (DARPA), July 1989.
8. J. Cadiz, B. Goldiez, and J. Thompson, Summary Report—The First Conference on Standards for the Interoperability of Defense Simulations, Institute for Simulation and Training of the University of Central Florida Report IST-CF-89-1 (Contract No. N61339-89-C-0043) 1989.
9. IEEE Standard for Information Technology, *Protocols for Distributed Interactive Simulation Applications,* IEEE Std 1278-1993, New York: IEEE, May 1993.
10. A. Katz, Event correlation for networked simulators, *J. Aircraft,* **32** (3): 515–519, 1995.
11. A. Katz, M. Sharma, and D. E. Wahrenberger, Advanced Dead Reckoning and Smoothing Algorithms, Prepared for US Army STRICOM, Contract No. N61339-91-D-0001, Architecture & Standards, Delivery Order No. 0035, CDRL A030, May 25, 1996.

Reading List

G. Burdea and P. Coiffet, *Virtual Reality Technology,* New York: Wiley, 1994.

W. E. Larsen, R. J. Randle, and L. N. Popish (eds.), *Vertical Flight Training,* NASA Reference Publication 1373, 1996.

D. F. McAllister, (ed.), *Stereo Computer Graphics and other True 3D Technologies,* Princeton, NJ: Princeton Univ. Press, 1993.

J. M. Rolfe and K. J. Staples, *Flight Simulation,* Cambridge: Cambridge Univ. Press, 1986, 1990.

R. G. Stanton, *Numerical Methods for Science and Engineering,* Englewood Cliffs, NJ: Prentice-Hall, 1961.

AMNON KATZ
University of Alabama

AGE TESTING. See INSULATION AGING TESTING.

AGGREGATE COMPUTATION. See STATISTICAL DATABASES.

AGING OF INSULATION. See INSULATION AGING MODELS.