

Binghamton University
Flight and Ground Vehicle Simulation Update

Real Time Computing

Roy Latham

**CGSD Corporation, 2723 Silver Rapids Rd
Valley Springs, CA 95252
tel: 209-920-3012 <http://www.cgsd.com>
roy@cgsd.com**

What is Real Time?

- Action times are predictable
- Acceptable lag from control action
- Motion must be synchronized and continuous
- Synchronized to the Simulator Host
- A concern for the visual system, a necessity for motion control
- Slower than real time is *interactive*

Continuous Motion

- Has nothing to do with the display refresh rate
- New images must be computed often enough for apparent continuity, usually 15Hz to 60Hz
- The requirement depends upon the vehicle dynamics: the higher the dynamics the faster the required update rate
- Moving objects must also have continuity
- The viewpoint of each image must correctly correspond to the time the image is shown

Minimum Frame Rate

- **Guaranteed minimum frame rate does not guarantee continuity**
- **There will appear to be a jump if a frame is dropped**
- **The sequence of frames does not correspond to equal event times**

Acceptable Lag

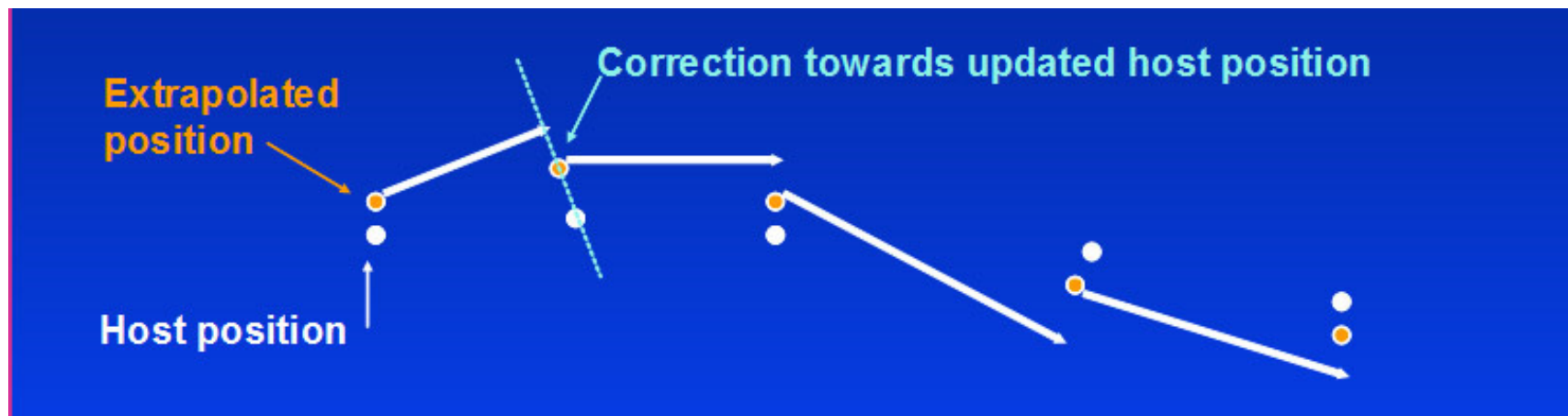
- Lag or *transport delay* is the extra time, due to the simulator, between when a control action is taken and when the first field of a new image is rendered showing the effect of the control action
- If lag is too long, the simulation will be unrealistic or even uncontrollable
- Acceptable lag depends on the application
- Usually lags less than 60 milliseconds can be made acceptable
- Lags of >250 msec are usually unacceptable for vehicles

Extrapolation

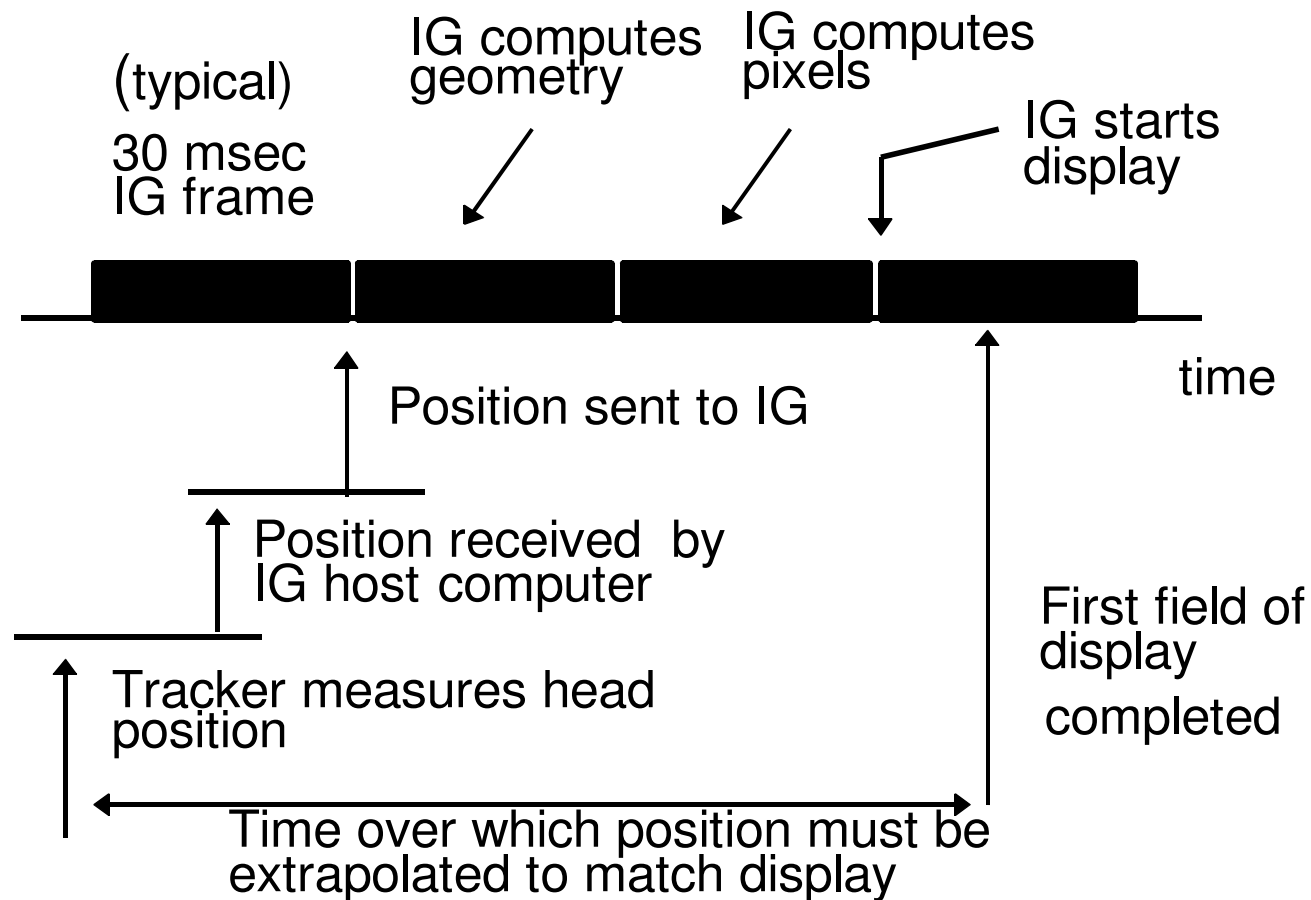
- Extrapolation can compensate for small lags
- $\text{Position} + \text{velocity} \times (\text{extrapolation time})$
- Sometimes the host vehicle model is not synchronized to the visual system, extrapolation compensates for the changing time differences
- The image generator always introduces a delay in making the image

Extrapolation Error

- Extrapolated position will be found to be in error, but cannot be suddenly corrected
- So, extrapolate the erroneous position forward
- Then extrapolate the corrected position forward
- Compute $(1 - W) \times P_{\text{old}} + W \times P_{\text{corr}}$



Example: VR System



Beware of hidden tracker lags

- Mechanical trackers have nearly zero lag
- No one makes a mechanical “head tracker” but a 3D digitizer is similar
- Microscribe G2X is 0.23 mm accuracy over 50 inches, \$7900



Synchronizing displays

- **One method is to genlock all channels**
 - Adds \$1K to graphics card cost
 - Frame time is the slowest channel
- **Alternative is extrapolate each channel to the correct time**
 - Use dedicated Ethernet with packet queue disabled to find time difference
 - Works with different and odd update rates

Real Time Software

- **Characteristic of true real time is that software execution times are predictable**
- **Allows events to be synchronized**
- **A hardware interrupt can be serviced in a guaranteed minimum amount of time**
- **System clock can be used for measured times**

Interrupts

- An interrupt is a hardware signal that stops current processing and transfer control to a new task
- The interrupt handler shuts off interrupts and saves the machine state, including the location of the pending instruction and any local data
- The highest priority interrupt is usually incrementing a counter, the software clock

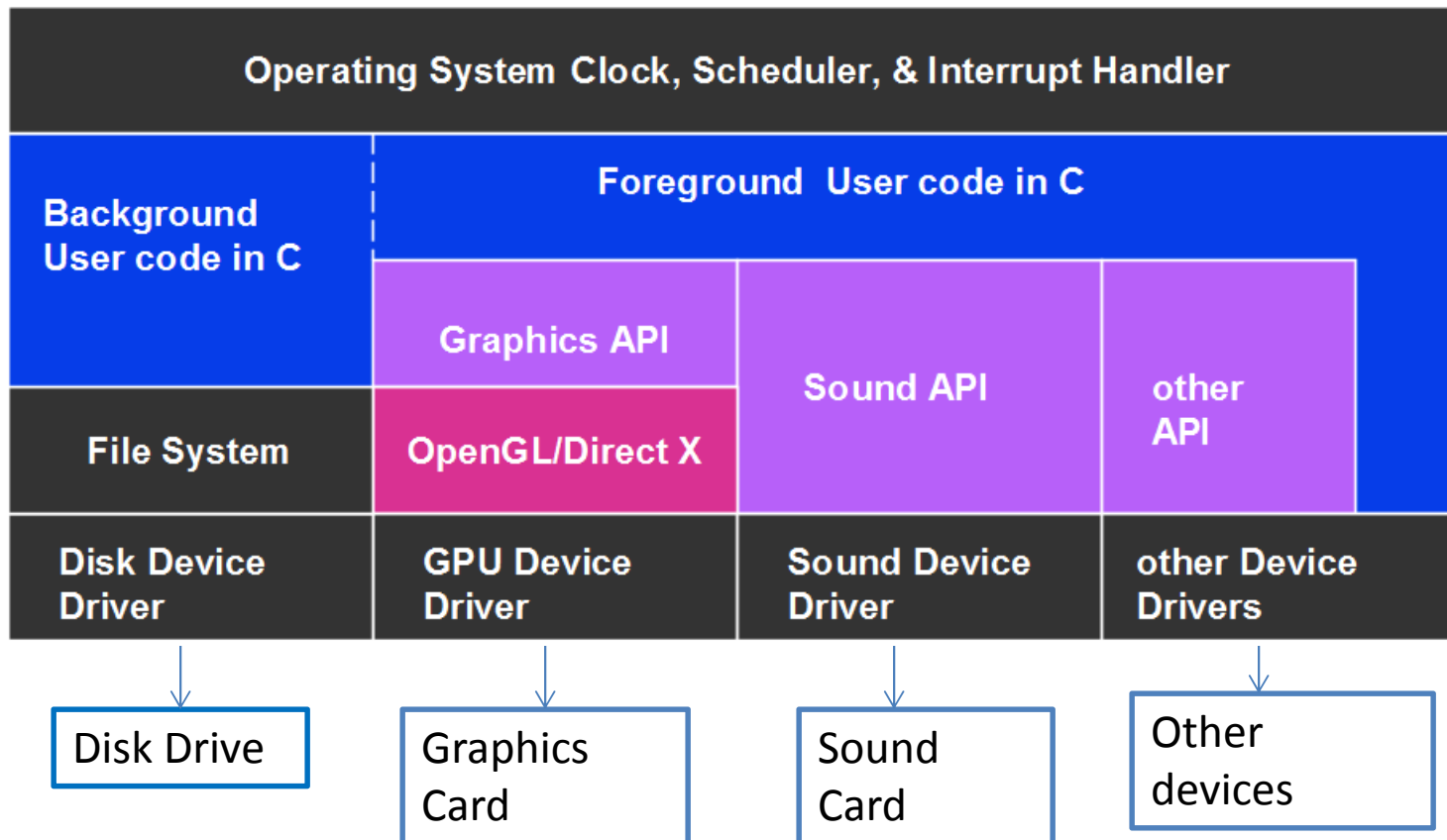
Graphics Interrupts

- The software must extrapolate position immediately before it starts the graphics processing for a frame: count 30 msec minus 0.1 msec
- If the SW misses the transfer of a new position, then the graphics engine will compute the image using the old data and two frames will be identical
- The computer can miss the interrupt if it's doing something else

The Problem with Disks

- **When a disk transfer is requested, it takes time for the disk to rotate into position and the head to seek the track**
- **If interrupts are off to service the hard disk, the graphics update can be missed**
- **Other devices create interrupts**
- **True real time allows the disk and other devices to be interrupted for the graphics update**

Simulator Software



Users of Real Time

- **Real time is a software niche**
- **Process control in manufacturing**
- **Control systems in aerospace**
- **Embedded systems for device control**
- **Games are usually not real time, just very fast**

Solutions

- **Traditionally, the whole computer system was designed for real time applications**
- **Now it is computers with real time operating systems:**
 - **Real time versions of UNIX**
 - **Proprietary operating systems**
 - **Extensions to Windows**

Device Drivers

- To work in a real time environment, device drivers must be interruptible
- This usually means writing custom device drivers
- On a PC, it always means someone must write custom device drivers
- Consequently, the vendors collection of device drivers is important
- Try to buy hardware that has RT driver support

Resources

- ***Dedicated Systems*** evaluation reports
<http://www.dedicated-systems.com/>
- Intime by TenAsys, for Windows and standalone
<http://www.tenasys.com/index.php/tenasys-products/intime-rtos-family/overview-rtos>
- RTX (embedded system with proprietary board) <http://www.keil.com/rl-arm/kernel.asp>

Reference Books

- *Real-time Operating Systems* by Jim Cooling (2013)
- *Embedded Linux Primer: A Practical Real-World Approach* by Christopher Hallinan (Prentice-Hall) 2nd ed.

Real Time Linux

- Real-Time Linux Foundation
<http://www.realtimelinuxfoundation.org/>
- RTLinuxPro: <http://www.fsmlabs.com/>
- LynxOS: <http://www.linuxworks.com/>
- Wind River Systems (major aerospace vendor)
<http://www.windriver.com/products/linux.html>

Proprietary Operating Systems

- VxWorks by Wind River Systems multi-platform, including PCs
<http://www.windriver.com/products/vxworks/>
- QNX real time OS
<http://www.qnx.com/products/neutrino-rtos/index.html>

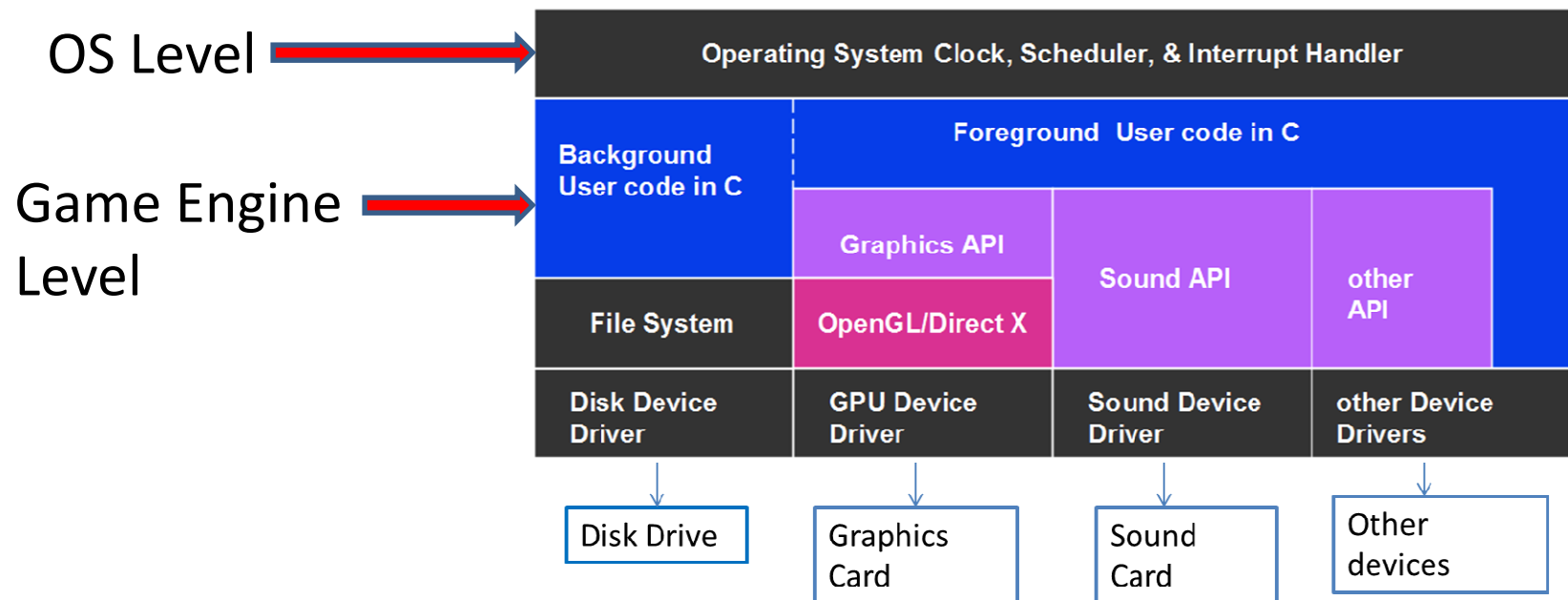
Extensions to Windows

- **Has advantages of the Windows[®] development environment**
- **Can be co-resident and interface via shared memory**
- **Windows cannot use windows devices while real time is in progress**
- **Example: QNX, InTime for Windows**

Do you really want real time?

- **Pseudo real time**
 - Load the whole database into memory so there are no disk interrupts
 - Solid State Memory instead of a disk
 - Use shared memory for paging
 - Make sure the frame rate is always 60 Hz
- **Tolerate occasional glitches**
 - Video game adept users tolerate glitches

How about game software?



Game software

- Game engines are tools that produce executable code that runs under an OS
- Pro: *Unreal Engine 4, Unity 5, Source 2, Frostbite 3 and Cry Engine 3.*
- Long list:
<http://www.slant.co/topics/1907/~game-engines-for-beginners-and-non-programmers>

Real Time Windows[tm]?

- Real time remains a niche, so it's unlikely to be put in a general purpose OS
- Games and multi-media are driving forces ... so maybe some day