Module –III
VIRTUAL ENVIRONMENT

Virtual environment

A **virtual environment** is a networked application that allows a user to interact with both the computing environment and the work of other users. Email, chat, web-based document sharing applications are all examples of virtual environments. Simply put, it is a networked common operating space. Once the fidelity of the virtual environment is such that it "creates a psychological state in which the individual perceives himself or herself as existing within the virtual environment" (Blascovich, 2002, pg 129) then the virtual environment (VE) had progressed into the realm of immersive virtual environments

Animation of Objects

The Dynamic Numbers

- Numerical Interpolation
- Linear Interpolation
- Non-Linear Interpolation

The Animation of Objects

- Linear Translation
- Non-Linear Translation

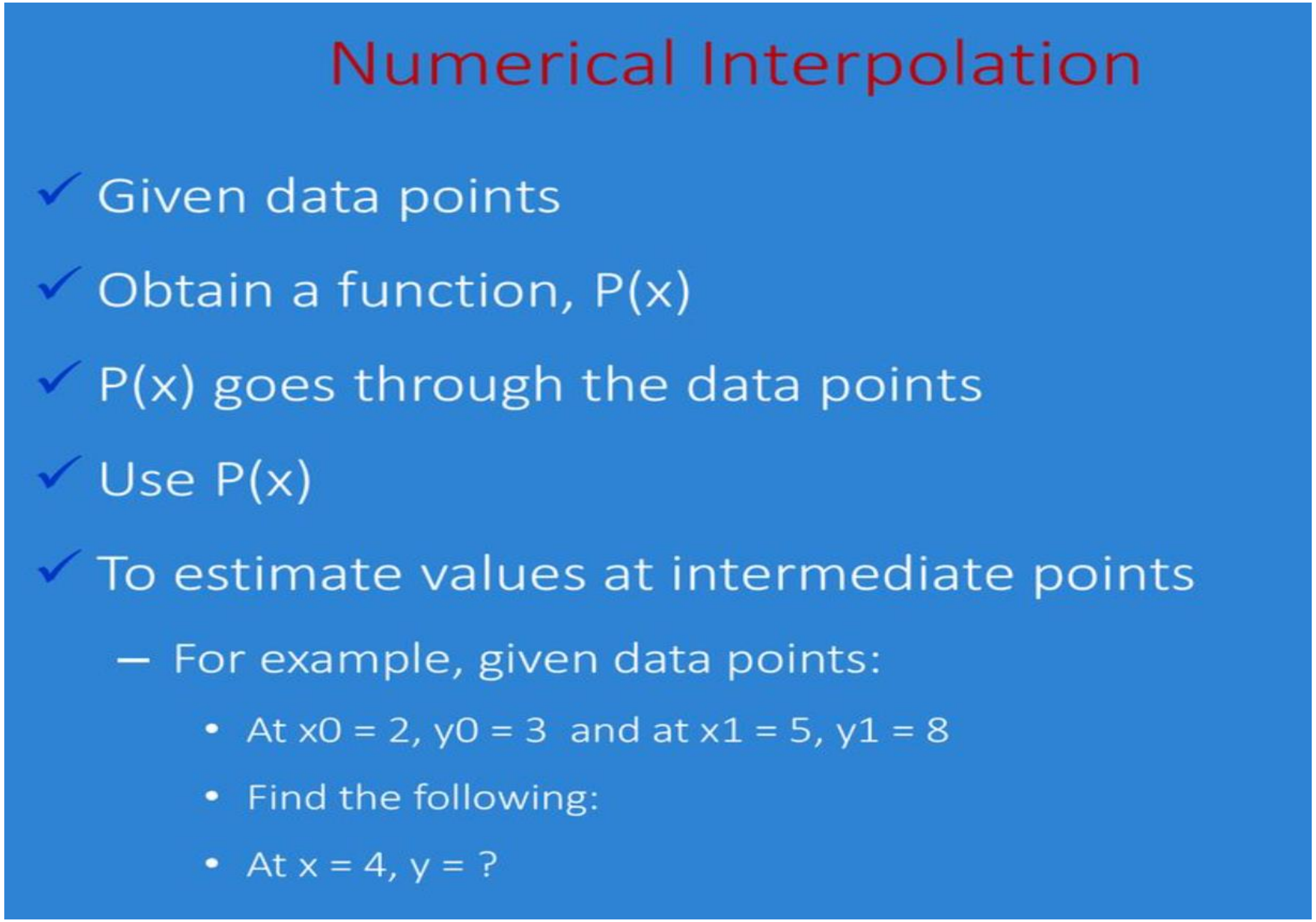
Shapes and Objects in between

The Dynamics of Numbers

- Virtual Environment – A Complex Numerical DB
- It changes one number to another
- Important: Numerical Envelope of the change
- Example: In animating the bouncing of a ball, it’s centroid is used to guide its motion
- Realistic motions are simulated by numerical interpolation
- Numerical Interpolation – Linear and Non-Linear

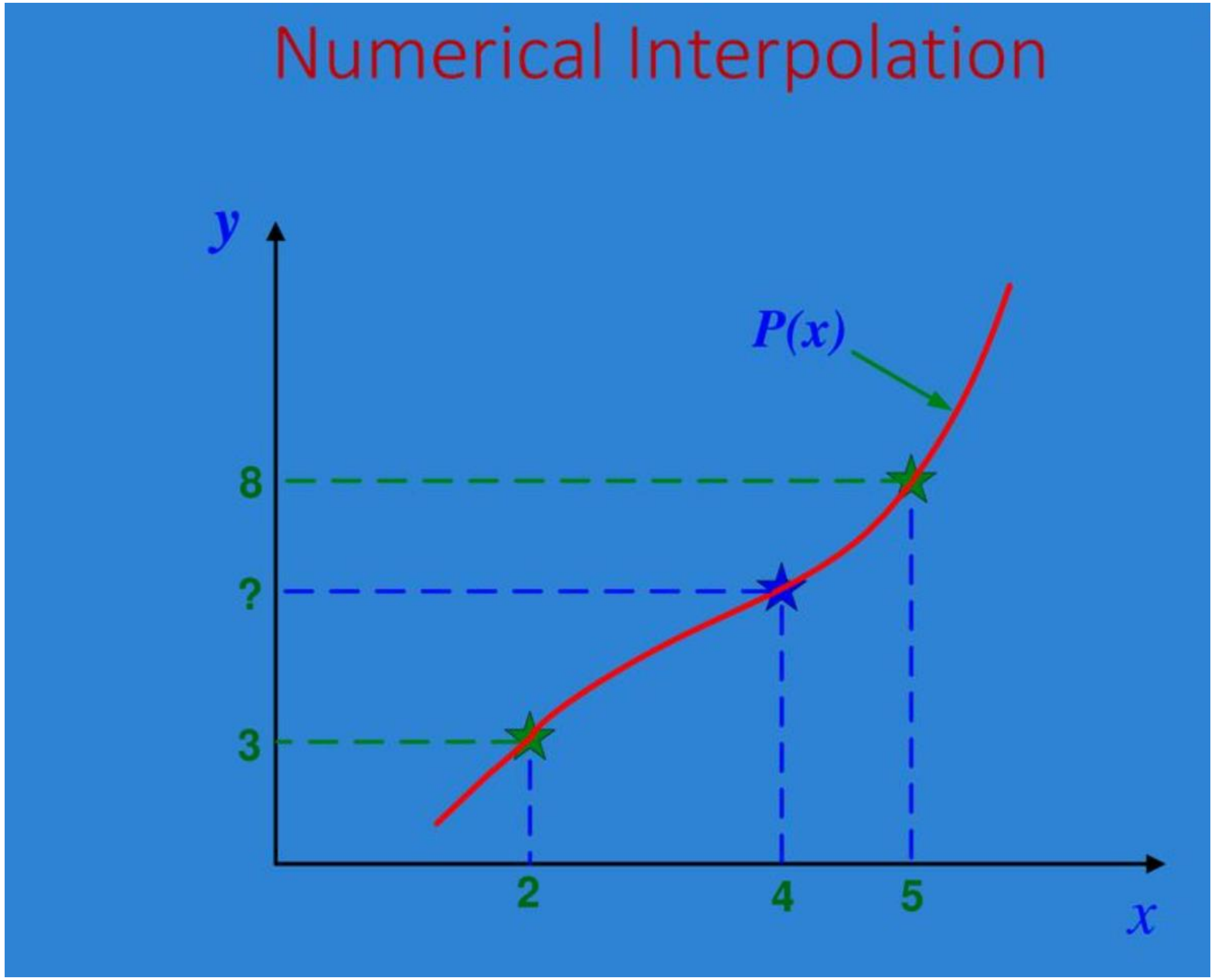
Numerical Interpolation

- Interpolation produces a function that matches the given data exactly.
- The function then can be utilized to approximate the data values at intermediate points
- It is also used to produce a function for which values are known only at discrete points, either from measurements or calculations.



Numerical Interpolation

- ✓ Given data points
- ✓ Obtain a function, P(x)
- ✓ P(x) goes through the data points
- ✓ Use P(x)
- ✓ To estimate values at intermediate points
  - For example, given data points:
    - At x0 = 2, y0 = 3 and at x1 = 5, y1 = 8
    - Find the following:
    - At x = 4, y = ?


Numerical Interpolation
y
P(x)
8
?
3
2
4
5
x

Linear Interpolation

- Linear Interpolation - Simplest interpolation method
- Apply LI(a sequence of points) ☐ A polygonal line where each straight line segment connects two consecutive points of the sequence
- Therefore, for every segment (P,Q)
- $P(x) = (1 - x)P + xQ$ where $x \in [0,1]$By varying x from 0 to 1, we get all the intermediate points between P and Q
- $P(x) = P$ for $x = 0$ and $P(x) = Q$ for $x = 1$.
- We get points on the line defined by P, Q, when $0 \le x \le 1$

Non-Linear Interpolation

- A linear interpolation ensures that equal steps in the parameter x give rise to equal steps in the interpolated values
- It is often required that equal steps in x give rise to unequal steps in the interpolated values
- We can achieve this using a variety of mathematical techniques
- For example, we could use trigonometric functions or polynomials to achieve this

The Animation of Objects

- Newton's Laws of Motion provide a useful framework to predict an object's behavior under dynamic conditions
- This scenario can be simulated within a Virtual Environment
- It can be achieved through a simple linear translation of objects

Uses of Translation

Modeling transformations

- build complex models by positioning simple components transform from object coordinates to world coordinates

Viewing transformations

- Viewing transformations placing the virtual camera in the world
- specifying transformation from world coordinates to camera coordinates

Animation

- vary transformations over time to create motion

Linear Translation

- Consider an object located at VE's origin and assigned a speed So across the XY plane
- To simulate its sliding movement, the x & z coordinates of the object must be modified (tnow – tprev)Vo
- The object's new velocity can be computed after it is bouncing off the boundary.

Non-Linear Translation

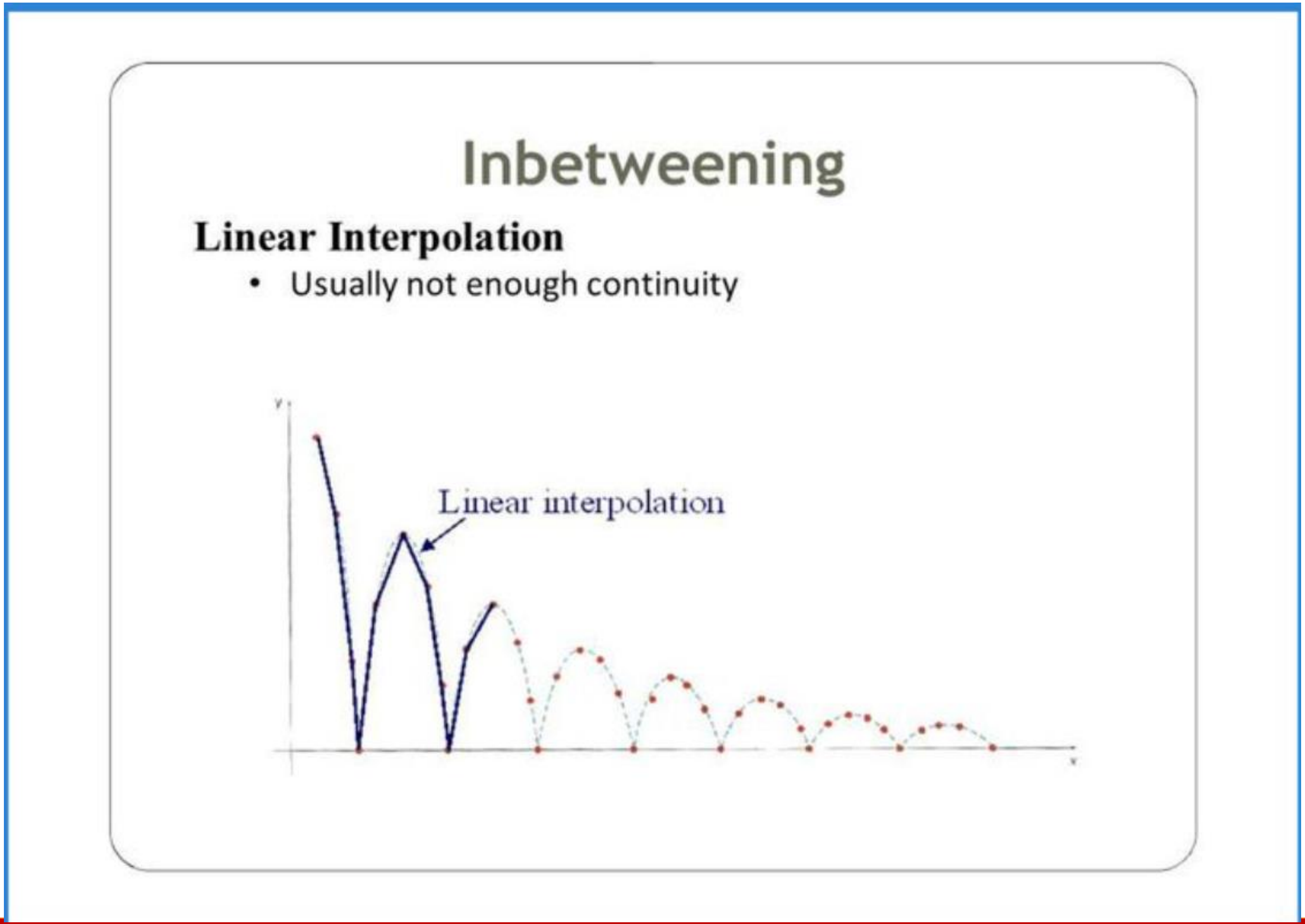
- Consider an object moving along the x-axis in 1s, pause momentarily and then returns to its original position in 2s
- The non-linear movement of the object can be simulated by computing the x-translation as a function of time
- At time t1, the translation begins. At time t2, i.e.,(t1+1) it pauses momentarily and at time t3, i.e., (t1+3), it comes to restt = T – t1 while t1 ≤ T ≤ t T – Current time t = (T – t1 – 1)/2 while t2 ≤ T ≤ t3 t- Control parameter

Shape and Object Inbetweening

- **Shape Inbetweening**

  It is a technique in the cartoon industry to speed up the process of creating art works
- The ‘inbetween’ objects are derived from two key images drawn by a skilled animator
- The images are called as Key Frames
- The key frames shows an object in two different positions
- The number and position of inbetweenig images determines the dynamics of final animation

Shape Inbetweening



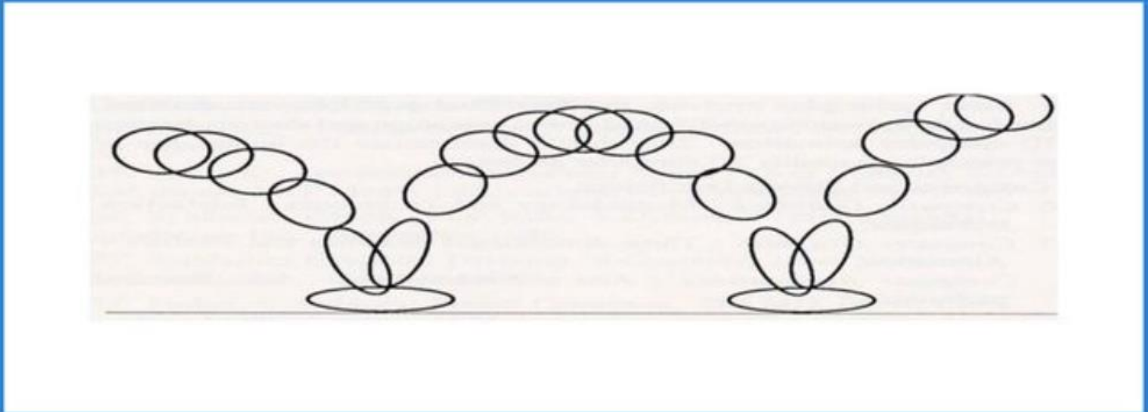
Inbetweening

Linear Interpolation

- Usually not enough continuity

Object Inbetweening

- By inbetweening the z-coordinate, the above technique can be applied to 3D contours
- We cannot expect to interpolate between two different complex objects and geometrically consistent
- Given suitable geometric definitions, it is possible to transform one object into another and create subtle animation



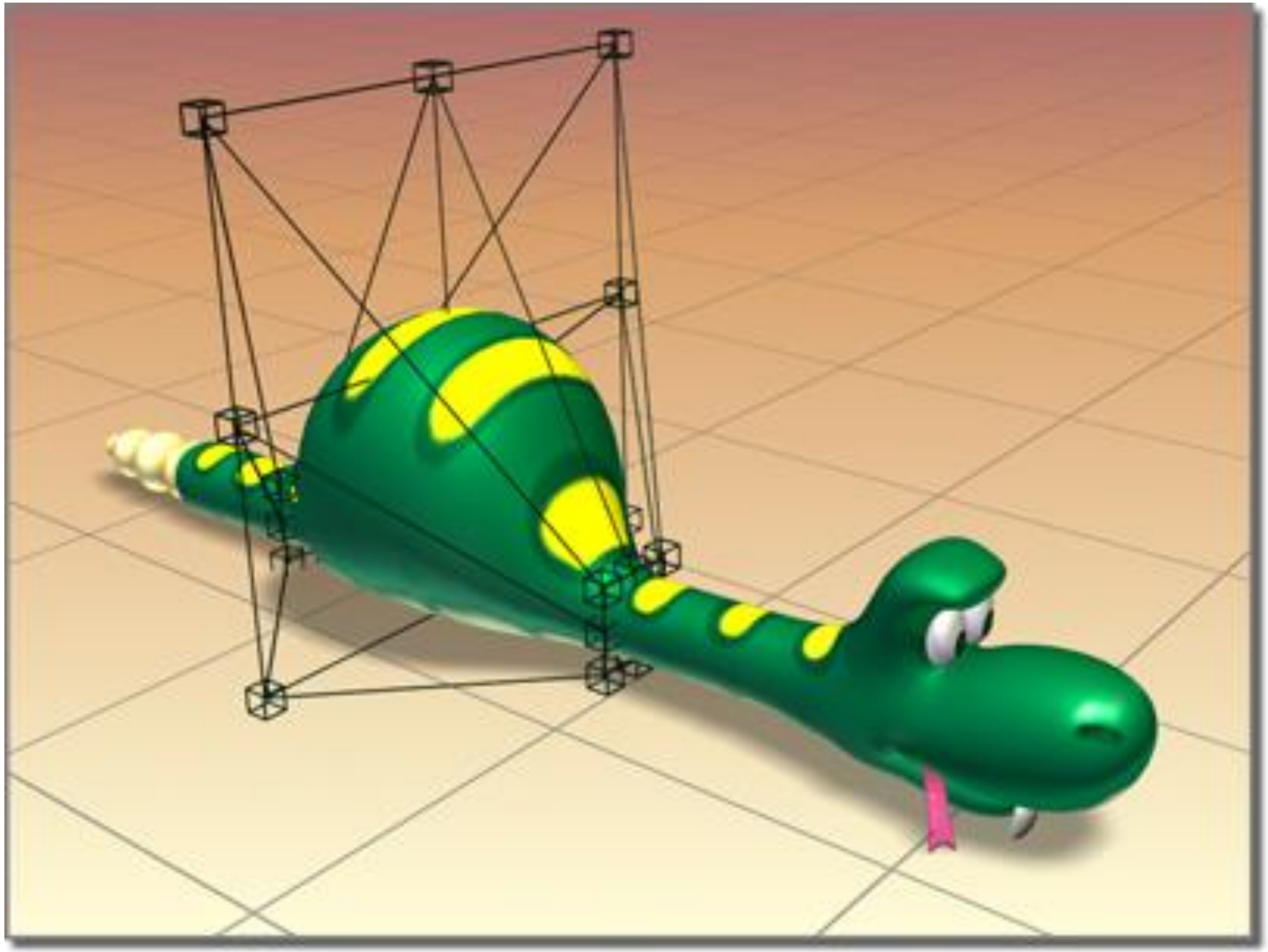
FFD (Free-Form Deformation) Modifiers

FFD stands for Free-Form Deformation. Its effect is used in computer animation for things like dancing cars and gas tanks. You can use it as well for modeling rounded shapes such as chairs and sculptures.

The FFD modifier surrounds the selected geometry with a lattice. By adjusting the control points of the lattice, you deform the enclosed geometry.

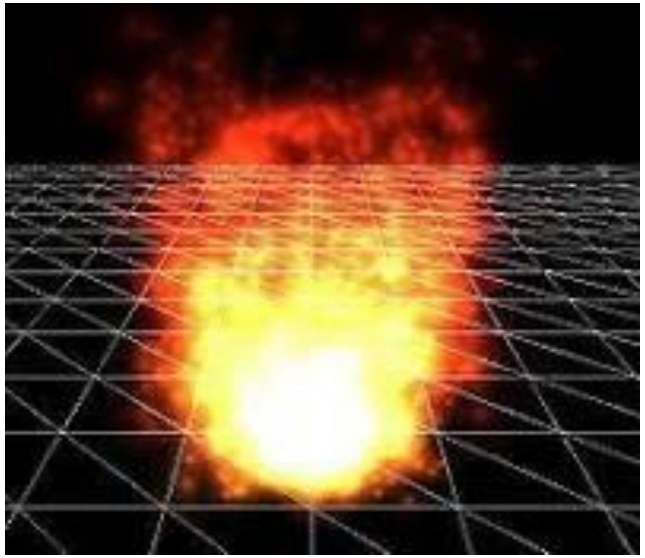
There are three FFD modifiers, each providing a different lattice resolution: 2x2x2, 3x3x3, and 4x4x4. The 3x3x3 modifier, for example, provides a lattice with three control points across each of its dimensions, resulting in nine on each side of the lattice.

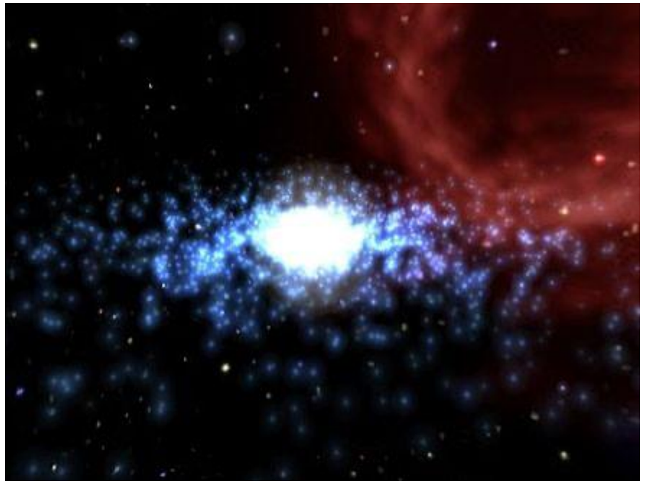
Also available are two more-configurable FFD modifiers; see FFD (Box/Cyl) modifier. These let you set any number of points in the lattice for greater flexibility in deforming the model.



A particle system

A **particle system** is a technique in game physics, motion graphics, and computer graphics that uses many minute sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena, which are otherwise very hard to reproduce with conventional rendering techniques - usually highly chaotic systems, natural phenomena, or processes caused by chemical reactions.





PHYSICAL SIMULATION

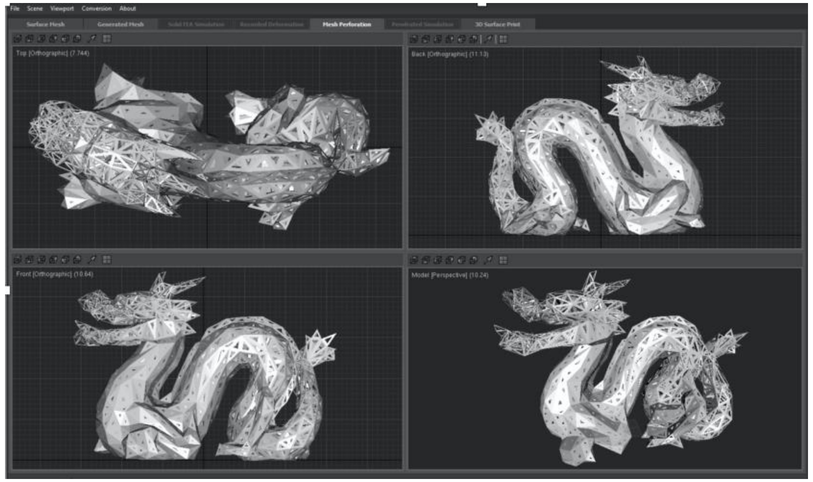
- Procedural Perforation for Elastic Structures
- Interactive Control of Deformable-object Animations through Control Metaphor Pattern Adherence
- Deformable Object Behaviour Reconstruction Derived through Simultaneous Geometric and Material Property Estimating

A **simulation** is an approximate imitation of the operation of a process or system; that represents its operation over time. Simulation is used in many contexts, such as simulation of technology for performance tuning or optimizing, safety engineering, testing, training, education, and video games. Often, computer experiments are used to study simulation models. Simulation is also used with scientific modelling of natural systems or human systems to gain insight into their functioning, as in economics. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. Simulation is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist.

Key issues in simulation include the acquisition of valid sources of information about the relevant selection of key characteristics and behaviors, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes. Procedures and protocols for model verification and validation are an ongoing field of academic study, refinement, research and development in simulations technology or practice, particularly in the work of computer simulation

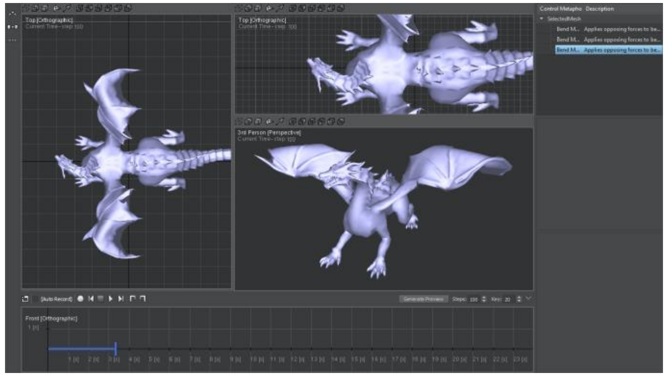
- Procedural Perforation for Elastic Structures

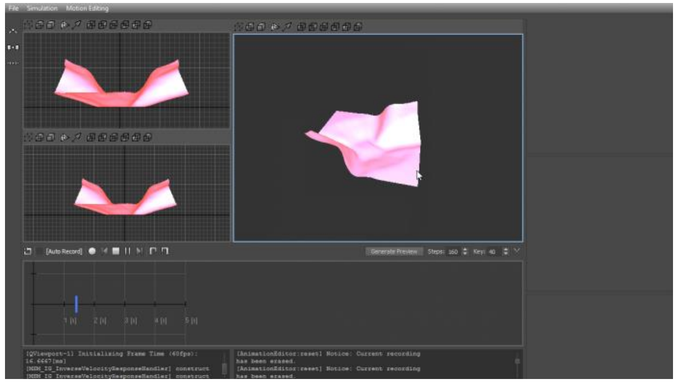
Procedural generation of elastic structures provides the fundamental basis for controlling and designing 3D printed deformable object behaviors. The automation through generative algorithms provides flexibility in how design and functionality can be seamlessly integrated into a cohesive process that generates 3D prints with variable elasticity. Generative deformation introduces an automated method for perforating existing volumetric structures, promoting simulated deformations, and integrating stress analysis into a cohesive pipeline model that can be used with existing consumer-level 3D printers with elastic material capabilities



Interactive Control of Deformable-object Animations through Control Metaphor Pattern Adherence

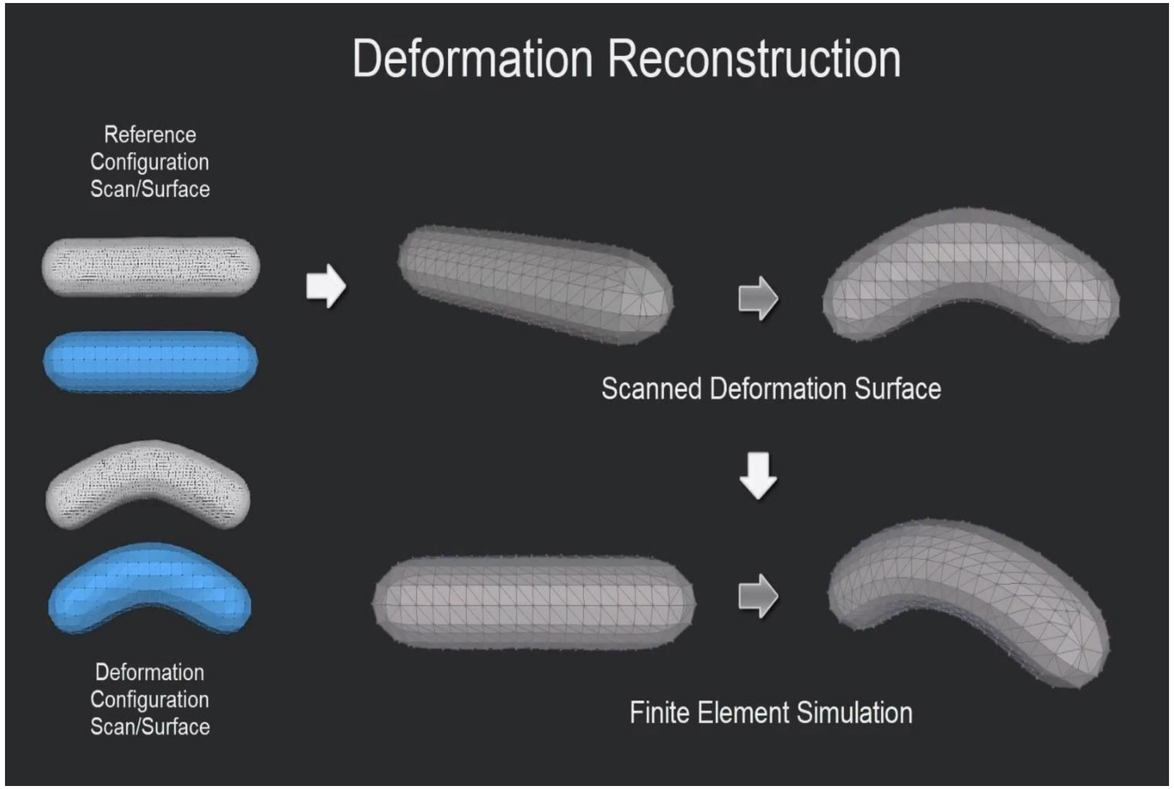
Adaptive and intuitive methodology for controlling the localized deformations of physically simulated objects using an intuitive pattern-based control interface. To maximize the interactive component presented in this approach we consolidate existing feedback mechanisms in deformable-body control techniques to provide intuitive editing metaphors for stretching, bending, twisting, and compressing simulated objects. The resulting movements created by these control metaphors are validated using imposed behavior evaluation and the effectiveness of this approach is demonstrated through interactively generated compound movements that introduce complex local deformations of objects in existing physical animations.





Deformable Object Behaviour Reconstruction Derived through Simultaneous Geometric and Material Property Estimation

The accurately reconstructing the deformation and surface characteristics of a scanned 3D model recorded in real-time within a Finite Element Model (FEM) simulation. Based on a sequence of generated surface deformations defining a reference animation, illustrate the ability to accurately replicate the deformation behavior of an object composed of an unknown homogeneous elastic material. We then formulate the procedural generation of the internal geometric structure and material parameterization required to achieve the recorded deformation behavior as a non-linear optimization problem. In this formulation the geometric distribution (quality) and density of tetrahedral components are simultaneously optimized with the elastic material parameters (Young's Modulus and Possion's ratio) of a procedurally generated FEM model to provide the optimal deformation behavior with respect to the recorded surface.



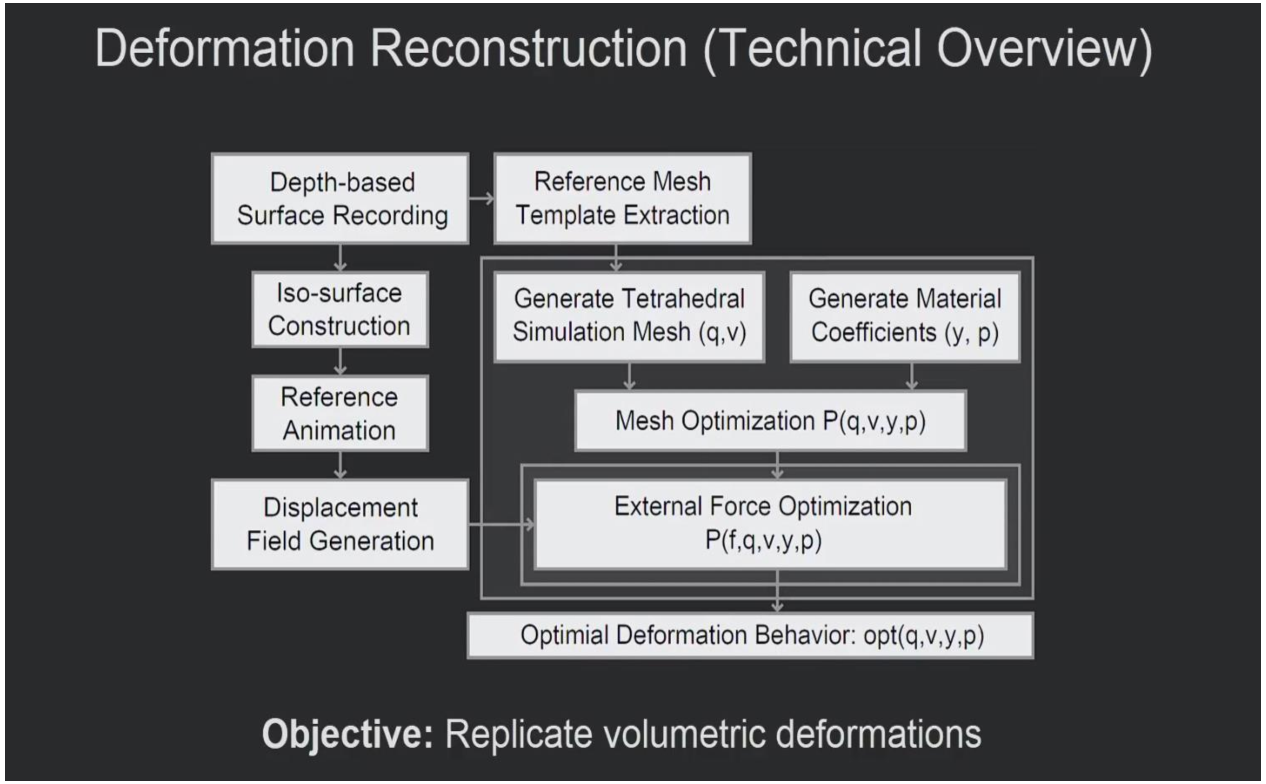
Deformation Reconstruction

Reference Configuration Scan/Surface

Scanned Deformation Surface

Deformation Configuration Scan/Surface

Finite Element Simulation



Deformation Reconstruction (Technical Overview)

Depth-based Surface Recording

Reference Mesh Template Extraction

Iso-surface Construction

Generate Tetrahedral Simulation Mesh (q,v)

Generate Material Coefficients (y, p)

Reference Animation

Mesh Optimization P(q,v,y,p)

Displacement Field Generation

External Force Optimization P(f,q,v,y,p)

Optimial Deformation Behavior: opt(q,v,y,p)

Objective: Replicate volumetric deformations

Objects falling in a graphical field

Falling objects form an interesting class of motion problems. For example, we can estimate the depth of a vertical mine shaft by dropping a rock into it and listening for the rock to hit the bottom. By applying the kinematics developed so far to falling objects, we can examine some interesting situations and learn much about gravity in the process.

Gravity

The most remarkable and unexpected fact about falling objects is that, if air resistance and friction are negligible, then in a given location all objects fall toward the center of Earth with the *same constant acceleration*, *independent of their mass*. This experimentally determined fact is unexpected, because we are so accustomed to the effects of air resistance and friction that we expect light objects to fall slower than heavy ones.

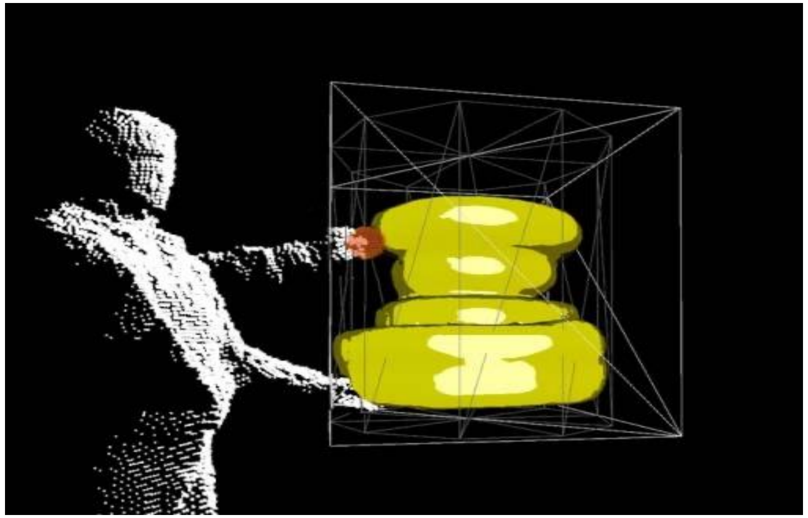
$g=9.80\ \text{m/s}^2$

In the real world, air resistance can cause a lighter object to fall slower than a heavier object of the same size. A tennis ball will reach the ground after a hard baseball dropped at the same time. (It might be difficult to observe the difference if the height is not large.) Air resistance opposes the motion of an object through the air, while friction between objects—such as between clothes and a laundry chute or between a stone and a pool into which it is dropped—also opposes motion between them. For the ideal situations of these first few chapters, an object *falling without air resistance or friction* is defined to be in **free-fall**.

The force of gravity causes objects to fall toward the center of Earth. The acceleration of free-falling objects is therefore called the **acceleration due to gravity**. The acceleration due to gravity is *constant*, which means we can apply the kinematics equations to any falling object where air resistance and friction are negligible. This opens a broad class of interesting situations to us. The acceleration due to gravity is so important that its magnitude is given its own symbol, gg. It is constant at any given location on Earth and has the average value

Rotating wheels

Is it possible to transfer real-world sculpting expertise to a virtual space? Through digital pottery, the answer is yes. Digital pottery is a collection of systems that makes it convenient to make pottery in 3D space. A natural & tangible user interface system that fluently connects users' real-world sculpting experience to virtual pottery making. Introducing a real spinning wheel, the physical concept of making pottery into 3D space without the burden of bridging the gaps between real & virtual worlds.

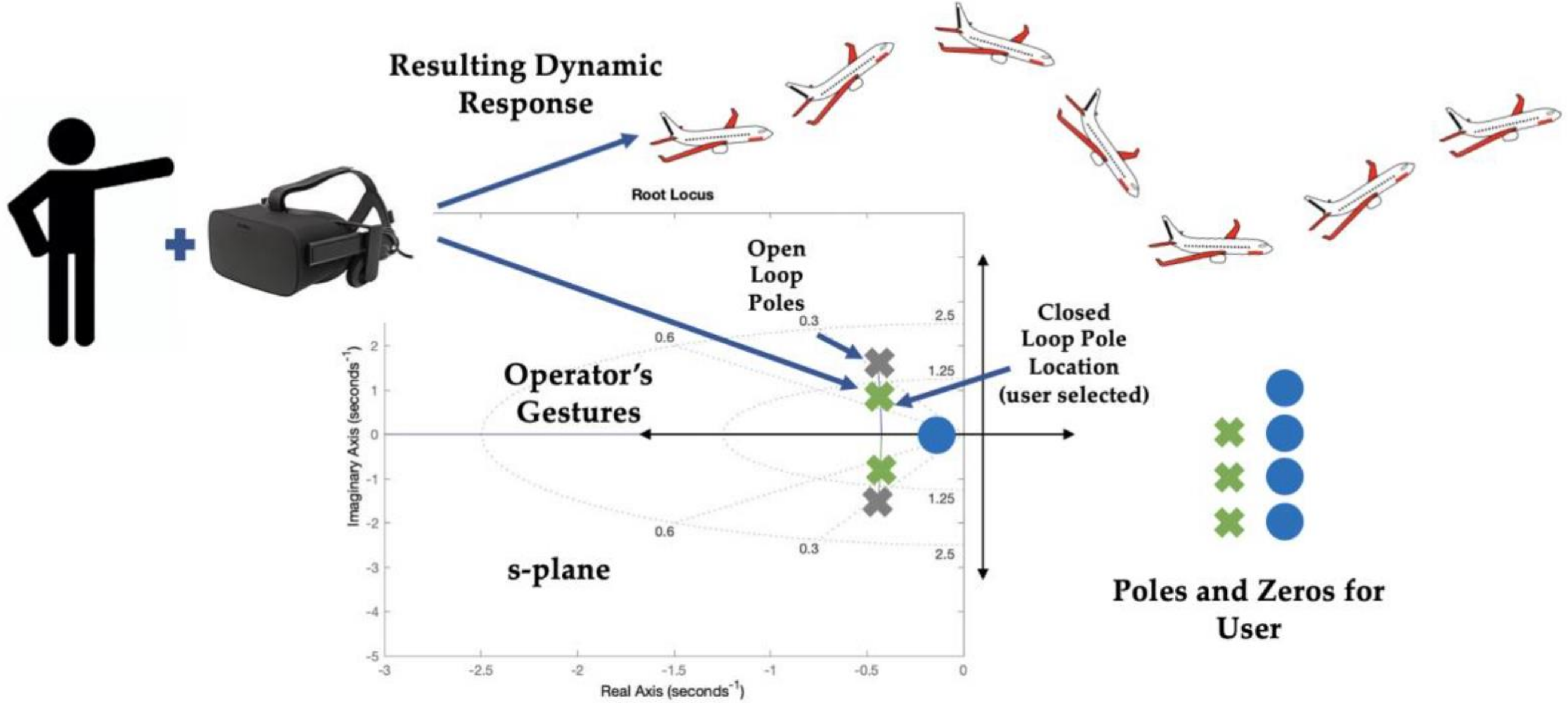


The current design of the virtual environment is built around the model of a fixed wing aircraft – the model is viewable by the user in a default viewing screen and the corresponding mathematical model saved in the background. In this effort, we are limiting our focus to one specific dynamic mode of the aircraft (short period); within the virtual environment, the user will be presented with a pulldown menu that gives them access to the *s-plane* widget. This renders the s-plane, with the real and imaginary axes, and displays the open loop poles and zeros of the current aircraft model.

This window will display the poles of the current aircraft configuration and give the user the option to place additional zeros and poles to the system. This window will also give the user the ability to tune various gain values of K for each controller component. During this process the user inputs will be fed through to various control computations running in the background and calculate the resulting system open-loop and closed-loop dynamics. The user can then exit the window after setting the controller conditions and run the VR simulation.

Flight dynamics of an aircraft

This simulation will begin with the aircraft moving forward at a constant speed and level altitude representing the steady state assumptions necessary for the small perturbation equations to hold true. The aircraft will then experience an elevator step input exciting the short period dynamics and producing a short-term system response. At this point in time the user will be able to reposition themselves around the aircraft by moving their body position relative to the aircraft in their headset, giving them the ability to experience the system response from multiple different perspectives. This simulation will run for a set amount of iterations before returning to the default viewing screen where the user can update the controller configuration. A simplified overview of the virtual environment and corresponding processes are represented in Figure below depicting the user headset interactions and resulting visual output.


Resulting Dynamic Response
Root Locus
Open Loop Poles
Closed Loop Pole Location (user selected)
Operator's Gestures
s-plane
Poles and Zeros for User
0.3
0.6
2.5
1.25
1.25
0.6
0.3
2.5
Imaginary Axis (seconds^{-1})
2
1
0
-1
-2
-3
-4
-5
-3
-2.5
-2
-1.5
-1
-0.5
0
Real Axis (seconds^{-1})

Aircraft Dynamics

The dynamics of a fixed wing aircraft can be described by a set of non-linear equations representing the translational and rotational motion of the aircraft, when subjected to external forces and moments. These equations of motion can be broken up into two independent sets describing the aircraft's lateral and longitudinal motion. For the purpose of this study, we limit our discussions to the longitudinal dynamics and linearize the aircraft dynamics around small perturbations about a wings level, steady state flight condition.

The longitudinal dynamics of a fixed wing aircraft can be further split into two second order dynamics – short period and phugoid dynamics. The system poles which represent the longitudinal stability of an aircraft for each of the two modes described above can be found below in Figure 2. The short period response can be characterized by a rapid change in angle of attack α and pitch attitude θ and has direct implications on the dynamic stability of the aircraft as well as the handling qualities and pilot workload.

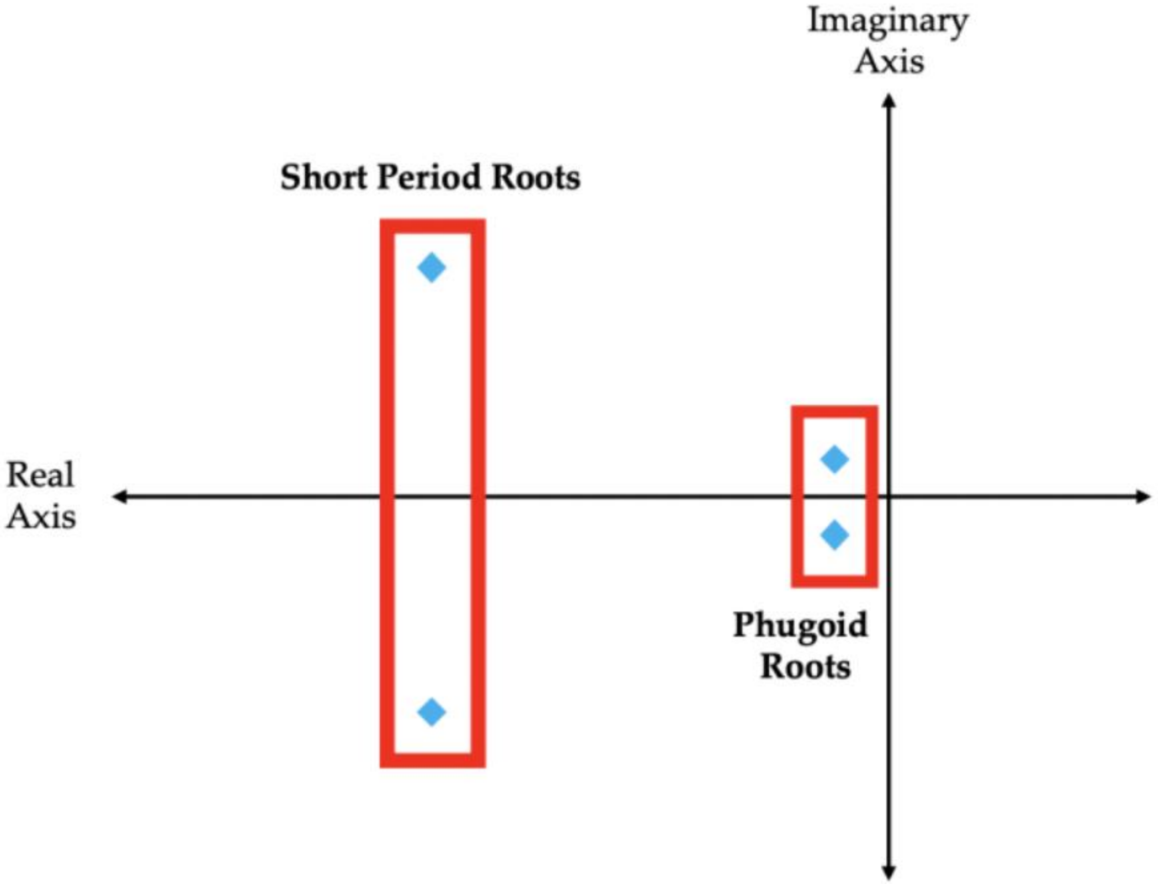


Figure 2: Pole locations describing the longitudinal dynamics of an aircraft

Module III
Revision