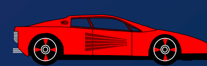
Hybrid Polynomial Prediction

- The remote host can dynamically choose the order of prediction polynomial
  - first-order or second-order?
- First-order
  - fewer computational operations
  - good when acceleration changes frequently or when acceleration is minimal
  - prediction can be more accurate without acceleration information



Position History-Based Dead Reckoning

- Chooses dynamically between first-order and second-order
- Evaluates the object’s motion over the three most recent position updates
- If acceleration is minimal or substantial, use first-order
  - threshold cut-off values for each entity
- The acceleration behaviour affects to the convergence algorithm selection

- Ignores instantaneous derivative information
  - update packets only contain the most recent position
  - estimate velocity and acceleration
- Reduces bandwidth requirement
- Improves prediction accuracy in many cases

Limitations of Derivative Polynomials

- Add more terms to the derivative polynomial—why not?
- With higher-order polynomials, more information have to be transmitted
- The computational complexity increases
  - each additional term requires few extra operations
- Sensitivity to errors
  - derivative information must be accurate
  - inaccurate values for the higher derivatives might actually make the prediction worse

$$p(t) = ½at^2 + vt + p$$

Limitations of Derivative Polynomials (cont'd)

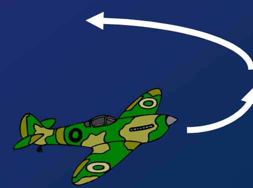
- Hard to get accurate instantaneous information
  - entity models typically contain velocity and acceleration
  - higher-order derivatives must be estimated or tracked
  - defining jerk (change in acceleration):
    - predict human behaviour
    - air resistance, muscle tension, collisions,…
  - values of higher-order derivatives tend to change more rapidly than lower-order derivatives

⇒High-order derivatives should generally be avoided

- The Law of Diminishing Returns
  - more effort typically provides progressively less impact on the overall effectiveness of a particular technique

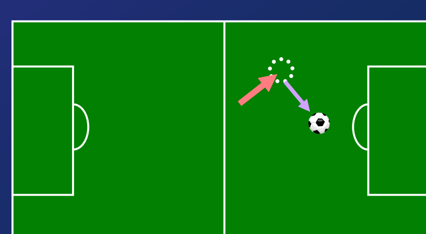
Object-Specialized Prediction

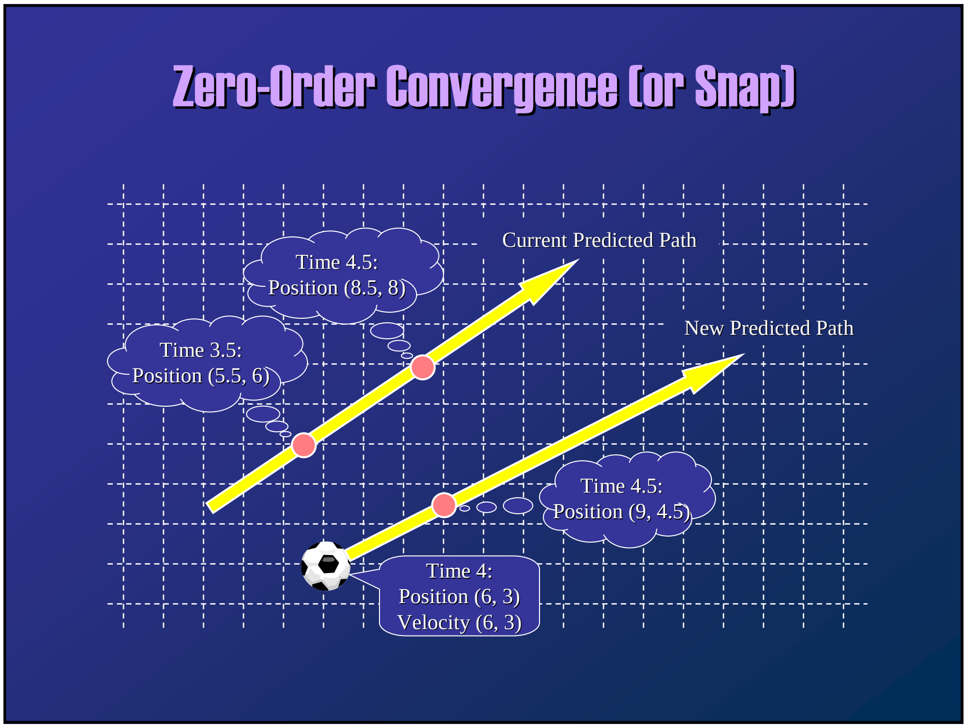
- Derivative polynomials do not take into account
  - what the entity is currently doing
  - what the entity is capable of doing
  - who is controlling the entity
- Managing a wide variety of dead reckoning protocols is expensive
- Aircraft making military flight manoeuvers
  - constant acceleration and instant velocity ⇒ position trajectory
  - the aeroplane's orientation angle
- All information does not need to be transmitted
  - dancing is relevant not the footwork, fire not the flames,…
- In general, precise behaviour would be nice but overall behaviour is enough

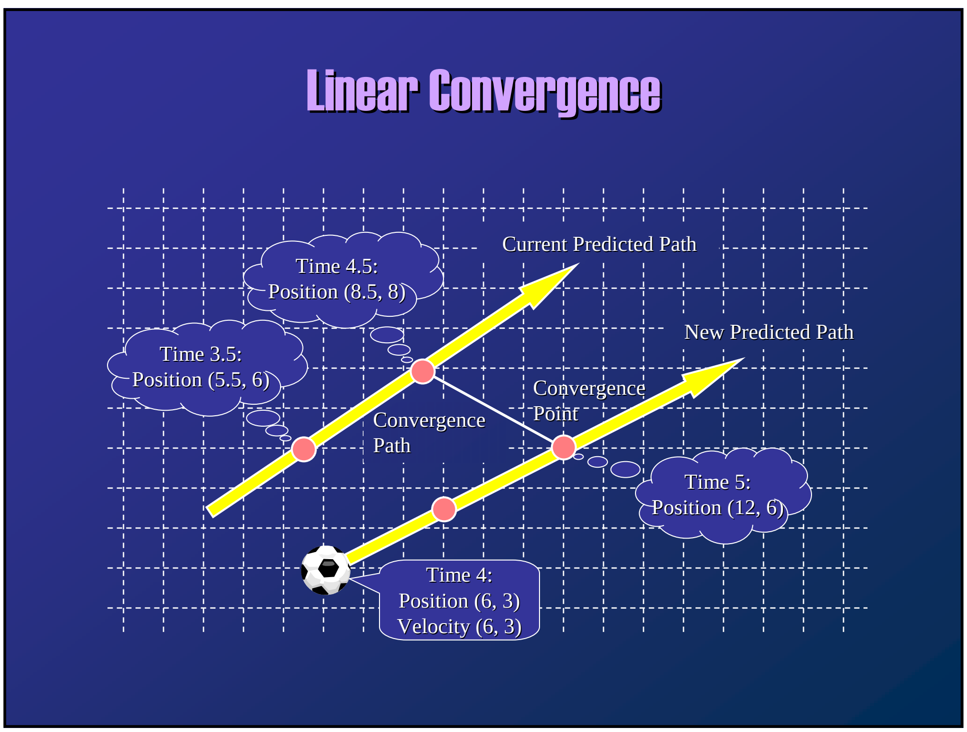


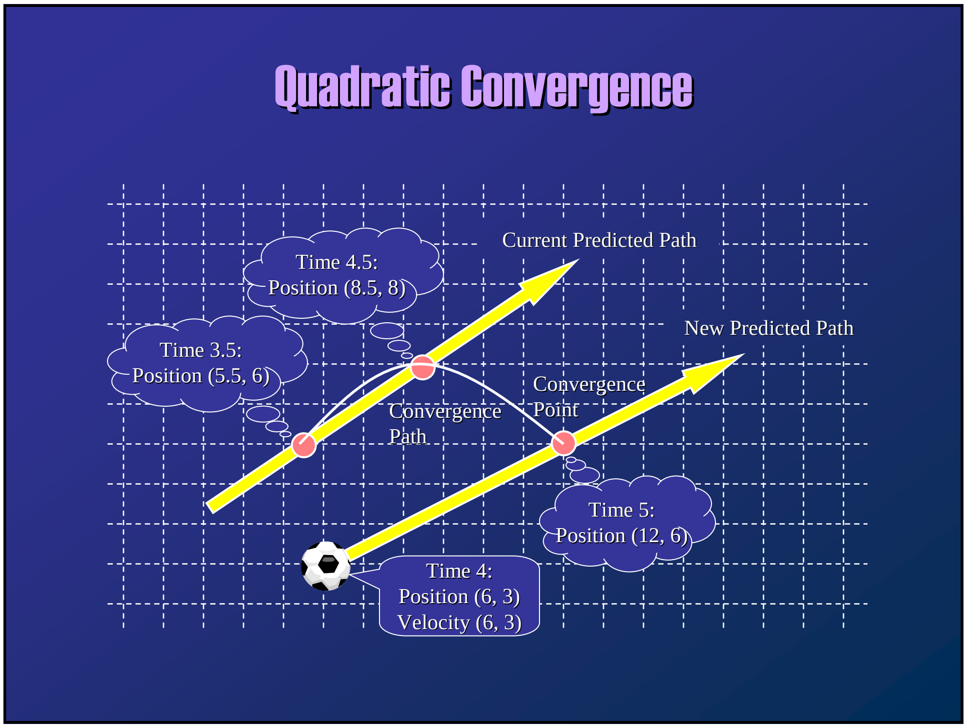
Convergence Algorithms

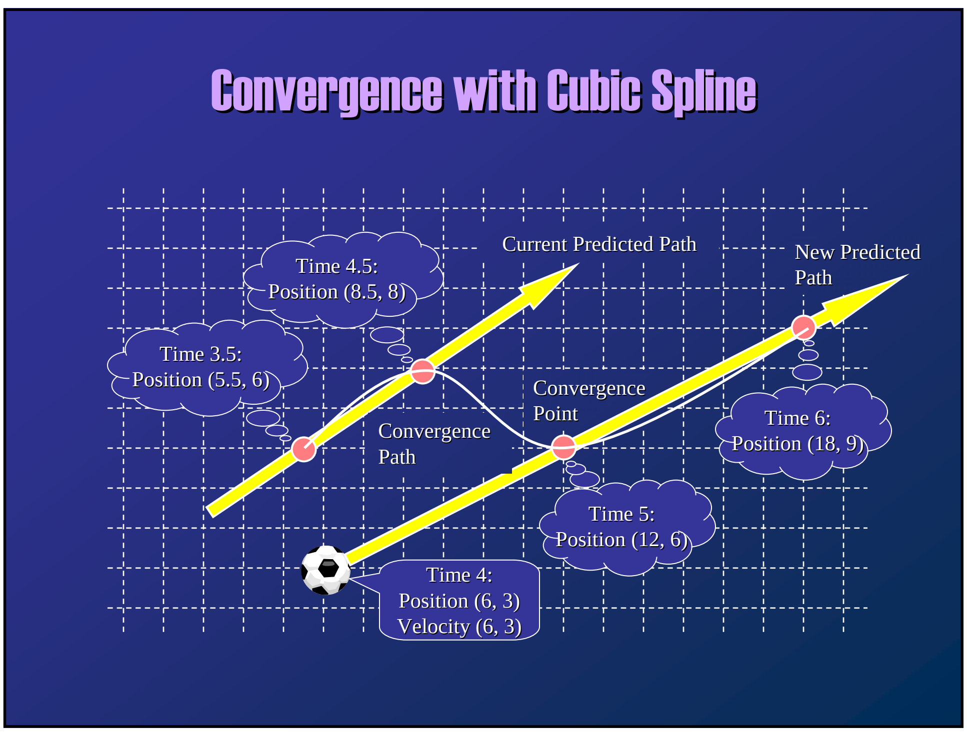
- Prediction estimates the future value of the shared state
- Convergence tells how to correct inexact prediction
- Correct predicted state quickly but without noticeable visual distortion




Zero-Order Convergence (or Snap)
Current Predicted Path
New Predicted Path
Time 4.5:
Position (8.5, 8)
Time 3.5:
Position (5.5, 6)
Time 4.5:
Position (9, 4.5)
Time 4:
Position (6, 3)
Velocity (6, 3)


Linear Convergence
Current Predicted Path
New Predicted Path
Time 4.5:
Position (8.5, 8)
Time 3.5:
Position (5.5, 6)
Convergence Point
Convergence Path
Time 5:
Position (12, 6)
Time 4:
Position (6, 3)
Velocity (6, 3)


Quadratic Convergence
Time 4.5:
Position (8.5, 8)
Current Predicted Path
Time 3.5:
Position (5.5, 6)
New Predicted Path
Convergence
Point
Convergence
Path
Time 5:
Position (12, 6)
Time 4:
Position (6, 3)
Velocity (6, 3)


Convergence with Cubic Spline
Time 4.5:
Position (8.5, 8)
Current Predicted Path
New Predicted
Path
Time 3.5:
Position (5.5, 6)
Convergence
Point
Convergence
Path
Time 6:
Position (18, 9)
Time 5:
Position (12, 6)
Time 4:
Position (6, 3)
Velocity (6, 3)

Nonregular Update Generation

- By taking advance of knowledge about the computations at remote host, the source host can reduce the required state update rate
- The source host can use the same prediction algorithm than the remote hosts
- Transmit updates only when there is a significant divergence between the actual position and the predicted position

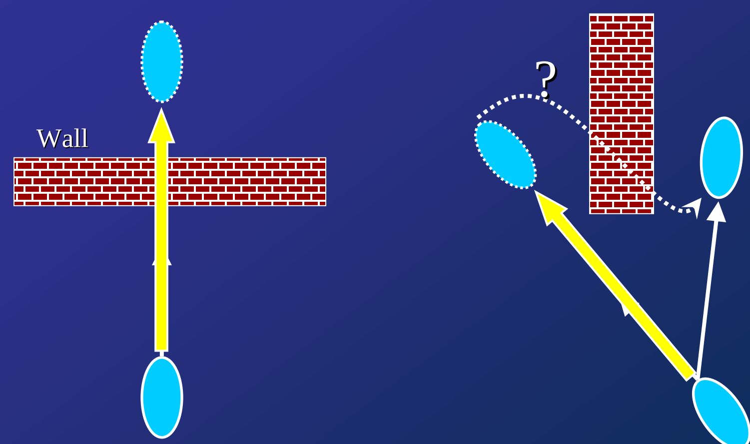
Advantages of Nonregular Transmissions

- Reduces update rates, if prediction algorithm is reasonable accurate
- Allows to make guarantees about the overall accuracy
- The source host can dynamically balance its network transmission resources
  - limited bandwidth $\Rightarrow$ increase error threshold
- Nonregular updates provide a way to dynamically adapt the consistency-throughput trade-off based on the changing consistency demands

Lack of Update Packets

- If the prediction algorithm is really good, or if the entity is not moving significantly, the source might never send any updates
- New participants never receive any initial state
- Recipients cannot tell the difference between receiving no updates because
  - the object's behaviour has not changed
  - the network has failed
  - the object has left the NVE
- Solution: timeout on packet transmissions

Environmental Effects



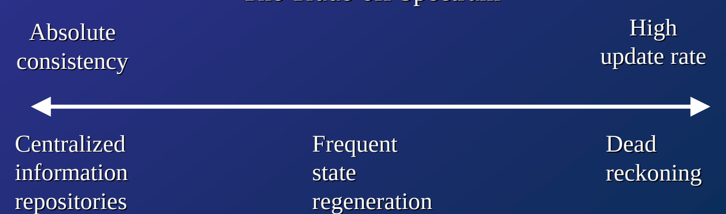
Dead Reckoning: Advantages and Drawbacks

- Reduces bandwidth requirements because updates can be transmitted at lower-than-frame-rate
- Because hosts receive updates about remote entities at a slower rate than local entities, receivers must use prediction and convergence to integrate remote and local entities
- Does not guarantee identical view for all participants
  - tolerate and adapt to potential differences
- Complex to develop, maintain, and evaluate
- Dead reckoning algorithms must often be customized for particular objects
- Are entities predictable?

Recapitulation: Managing Dynamic Shared State

- Consistency-throughput trade-off
- Centralized information repositories
- Frequent state regeneration
- Dead reckoning

The Trade-off Spectrum



§5 System Design

- Model-View-Controller
- Threads
  - single
  - multiple
- Important subsystems
  - real-time rendering
    - polygon culling
    - level-of-detail processing
  - real-time collision detection and response
  - computational resource management

Model-View-Controller