

CS 277 - Experimental Haptics

Lecture 2

“Haptic rendering, force fields”



Outline

- **Defining the problem**
- **Architecture of a Haptic Interface**
- **Virtual Walls and Potential Fields**
- **Issues with Potential Fields**
 - **Corner cases**
 - **Limitations of Virtual Environments**
- **Time is of the essence**

Outline

- **Defining the problem**
- Architecture of a Haptic Interface
- Virtual Walls and Potential Fields
- Issues with Potential Fields
 - Corner cases
 - Limitations of Virtual Environments
- Time is of the essence

Kinesthetic vs. Tactile Feedback

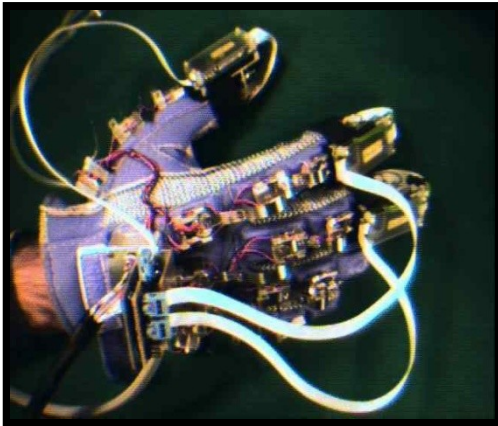
- **Humans perceive contact through sensors (mechanoreceptors) located in their skin, joints, tendons, and muscles.**
- **Simplistic distinction:**
 - **Tactile sense**
 - **Kinesthetic sense**

Tactile Devices

- **historically harder to implement due to**
 - **high spatial actuator resolution**



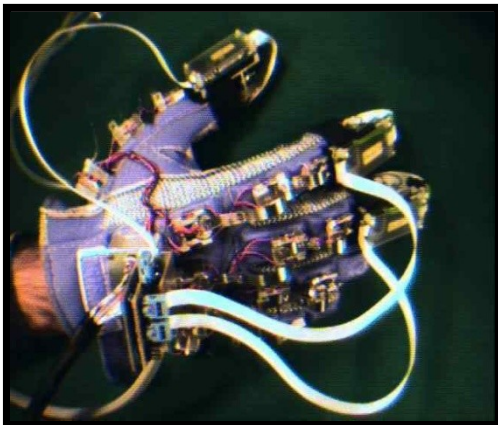
Kinesthetic Devices



OR



Kinesthetic Devices



OR



Impedance vs. Admittance devices

- **Impedance** haptic devices
 - Sense a position
 - Produce a force



- **Admittance** haptic devices
 - Sense a force
 - Produce a position



Impedance vs. Admittance devices

- **Impedance** haptic devices

- are cheaper to build
- don't need force sensors
- back drivable



- **Admittance** haptic devices

- require the use of a force sensor
- more expensive to produce
- provide higher range of forces



What Devices will we focus on?

- **Kinesthetic devices**
- **Impedance devices**

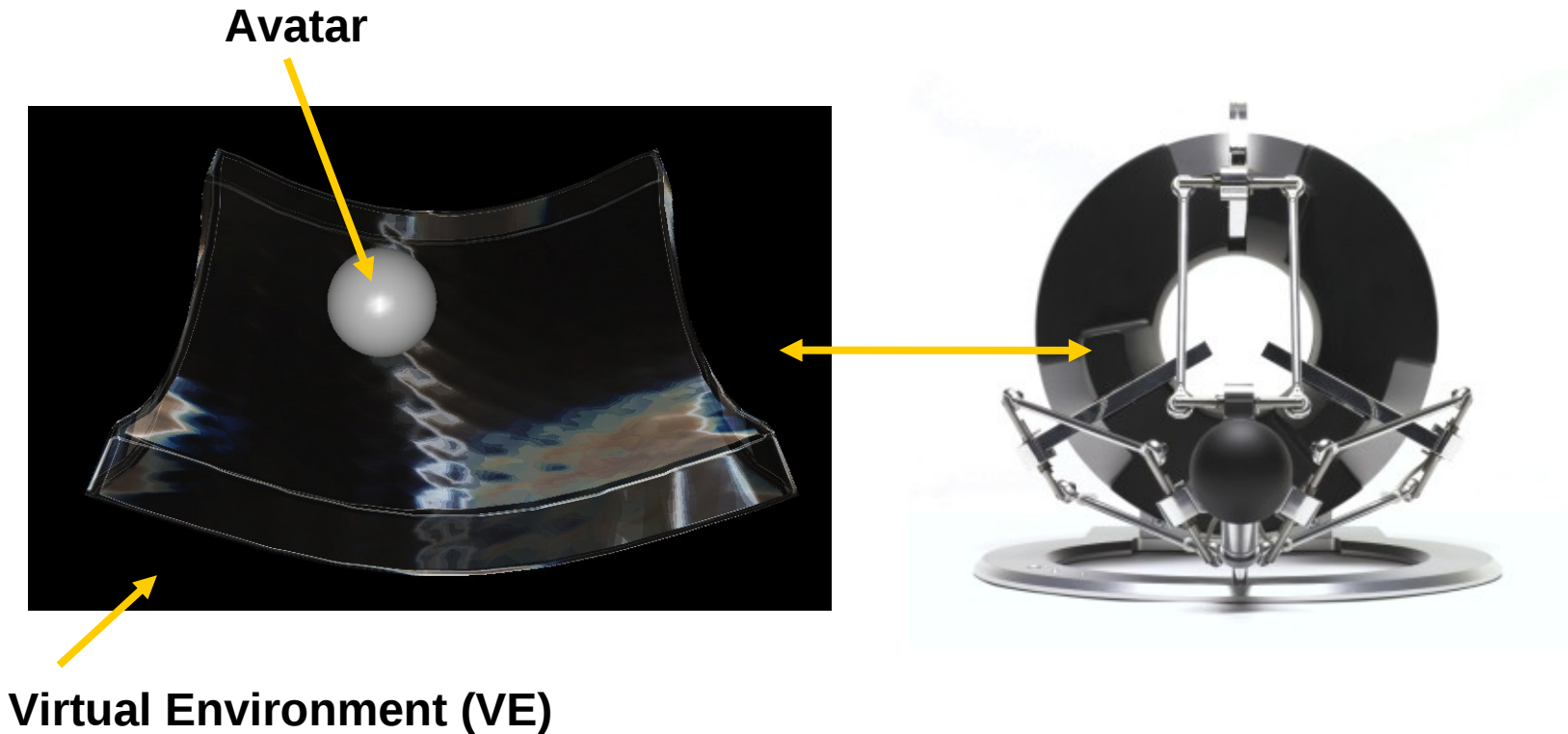


Outline

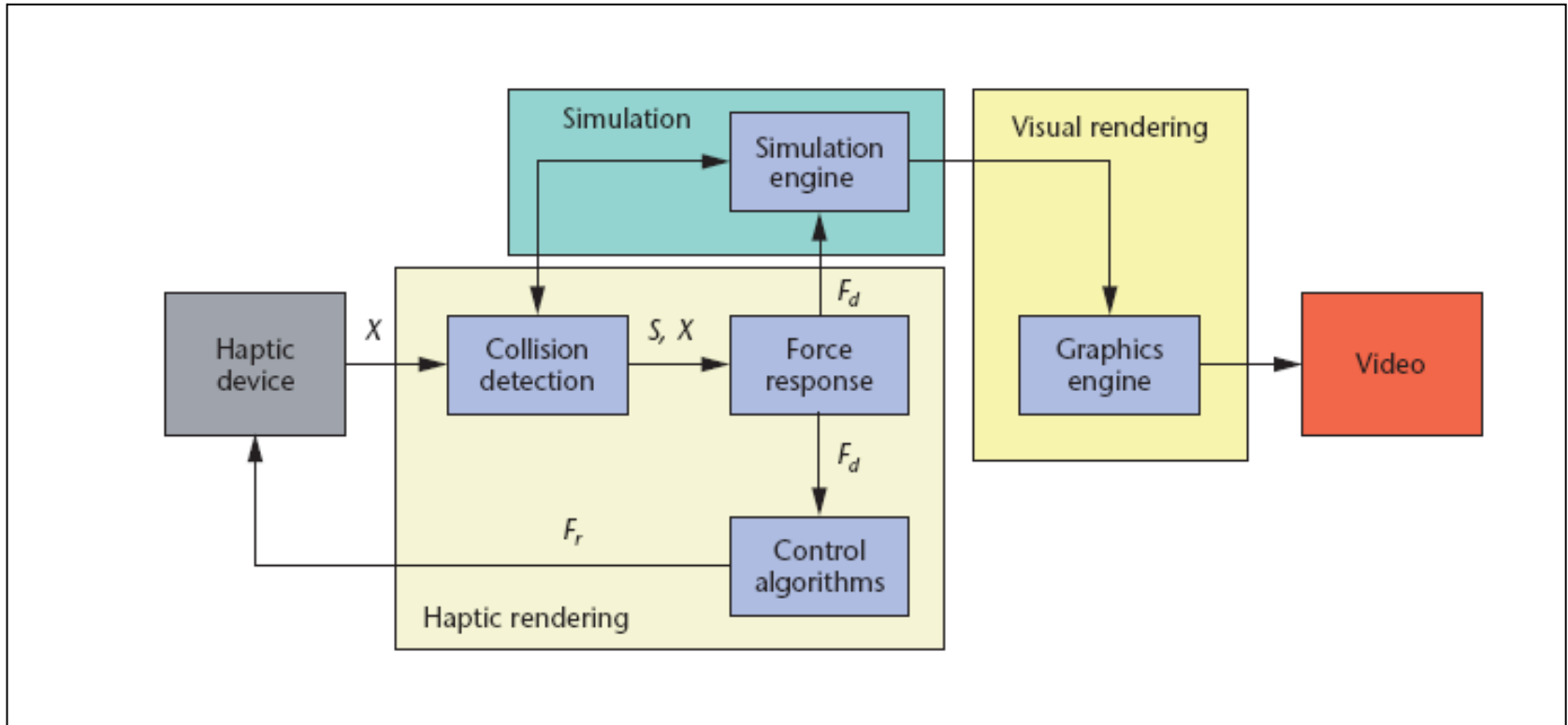
- Defining the problem
- **Architecture of a Haptic Interface**
- Virtual Walls and Potential Fields
- Issues with Potential Fields
 - Corner cases
 - Limitations of Virtual Environments
- Time is of the essence

Haptic Interface Architecture

- How does a basic haptic interface work?

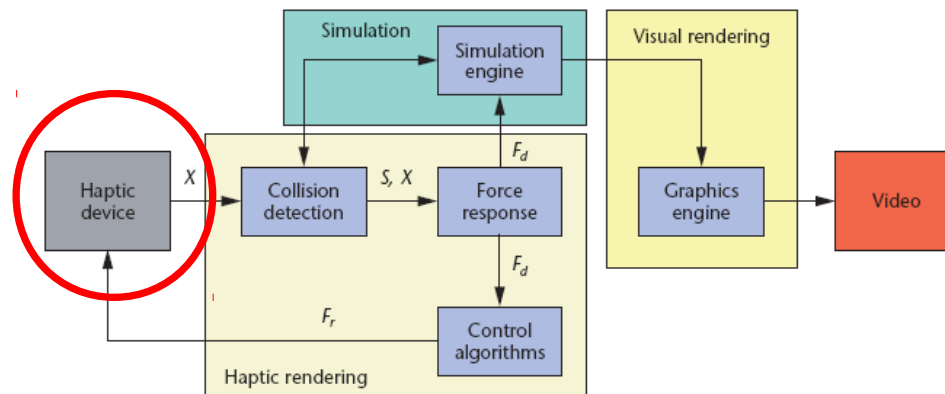


Haptic Interface Architecture



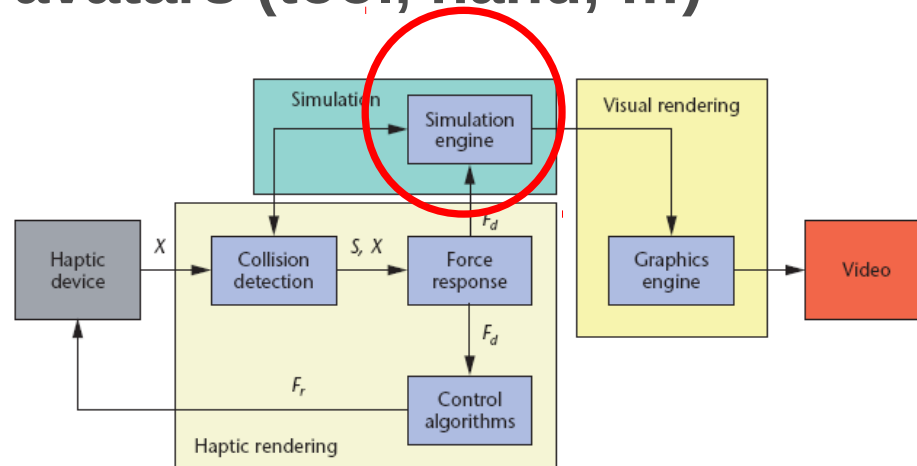
Haptic Interface Architecture

- Use haptic device to physically interact with the VE
 - optical encoders measure position of end effector
 - actuators apply forces to the user
 - haptic rendering algorithms compute such forces given the new positions



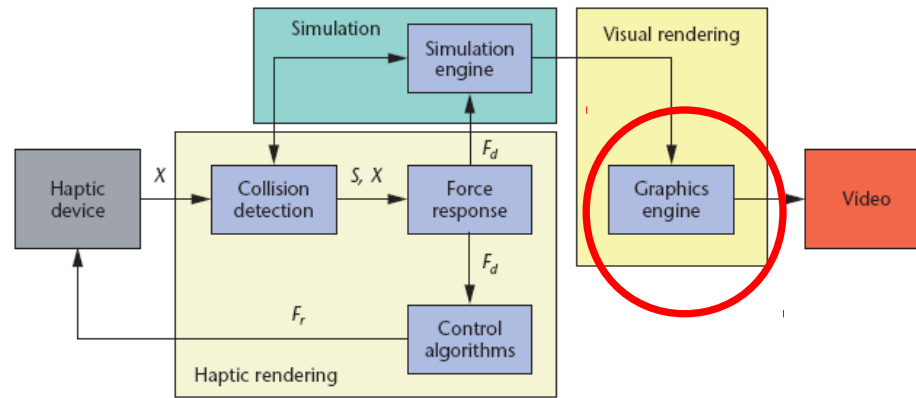
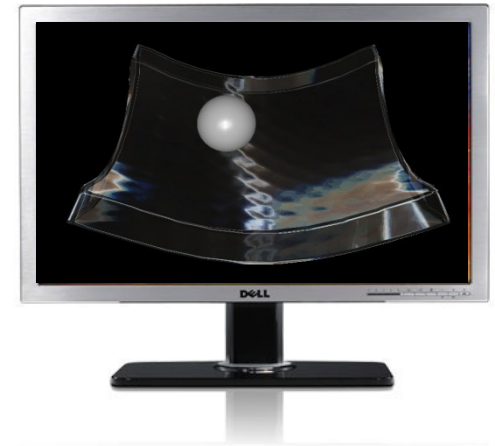
Haptic Interface Architecture

- **Assumption: haptically enabled Virtual Environments (VE)**
- **Simulation Engine computes state of VE**
 - virtual objects
 - user's avatars (tool, hand, ...)



Haptic Interface Architecture

- **Graphics Engine**
 - Given a VE, render it on my display
 - this class is **NOT** about graphics
 - **chai3d**: makes it easy to couple haptics and graphics

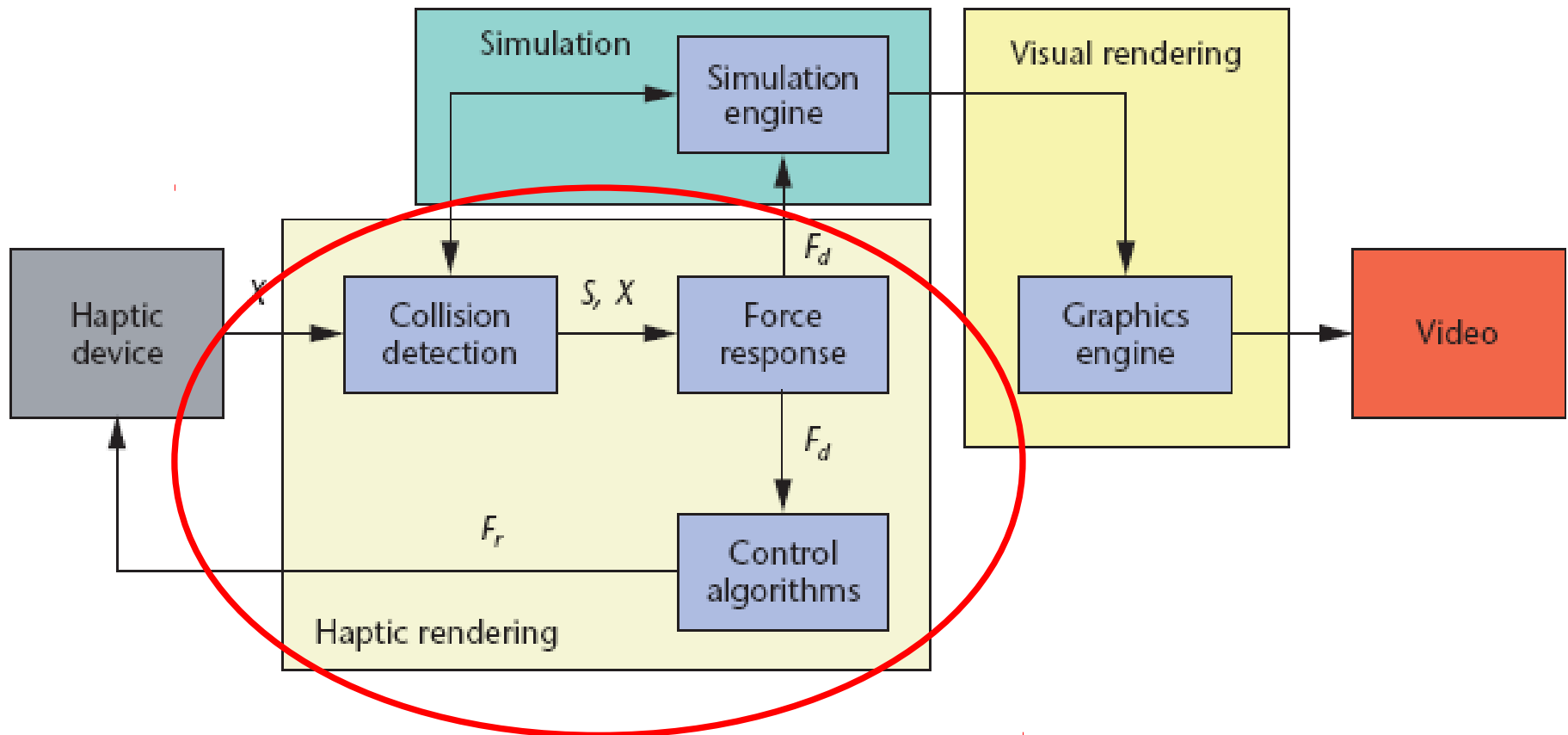


What Devices will we focus on?

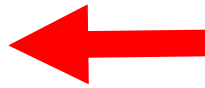
- **Recap:**
 - **kinesthetic devices,**
 - i.e. most likely tool-mediated interaction
 - **impedance devices,**
 - i.e. simulating virtual objects as impedances
 - **3 DOF actuated devices, 3 or 6 DOF sensed**



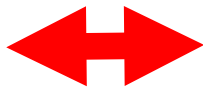
Haptic Interface Architecture: Haptic Rendering



Haptic Rendering: Parallels to Visual Rendering



**Visual
Rendering**

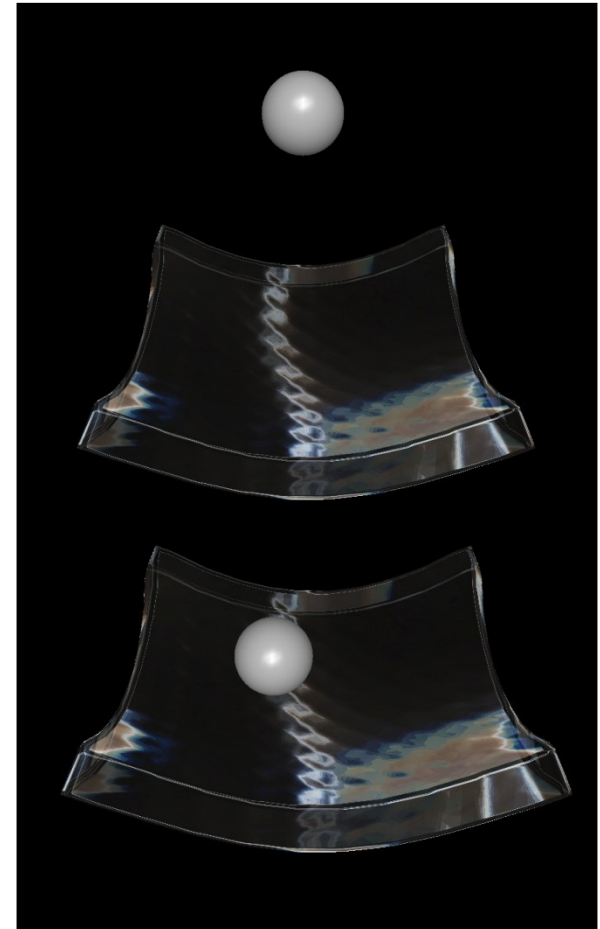
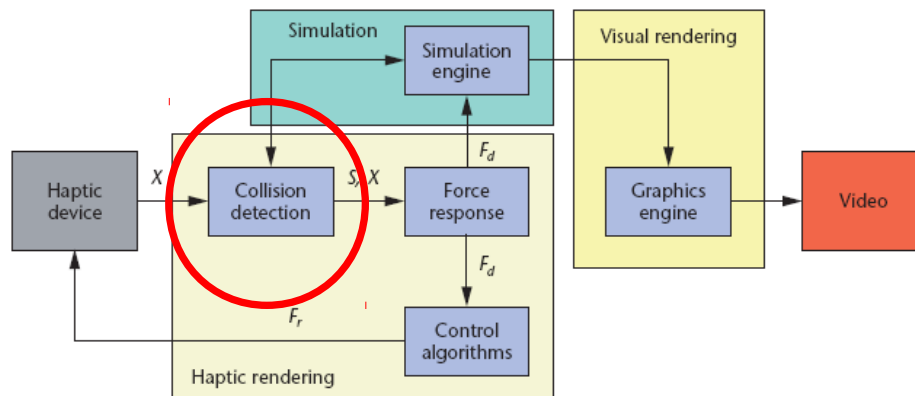


**Haptic
Rendering**



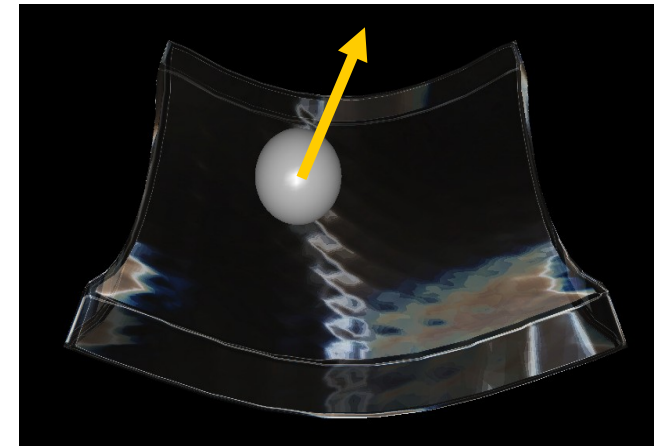
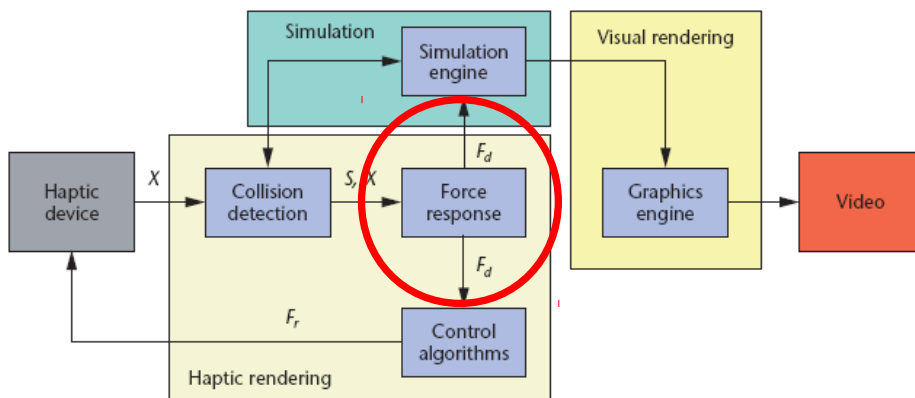
Haptic Interface Architecture

- **Collision detection**
 - Given a new position (x,y,z) from the device does my avatar collide with any object?
 - If so, what part of which object?



Haptic Interface Architecture

- **Force Response algorithms**
 - Given collided object compute reaction force (F_x , F_y , F_z) such that
 - contact “feels realistic”
 - contact “looks realistic”

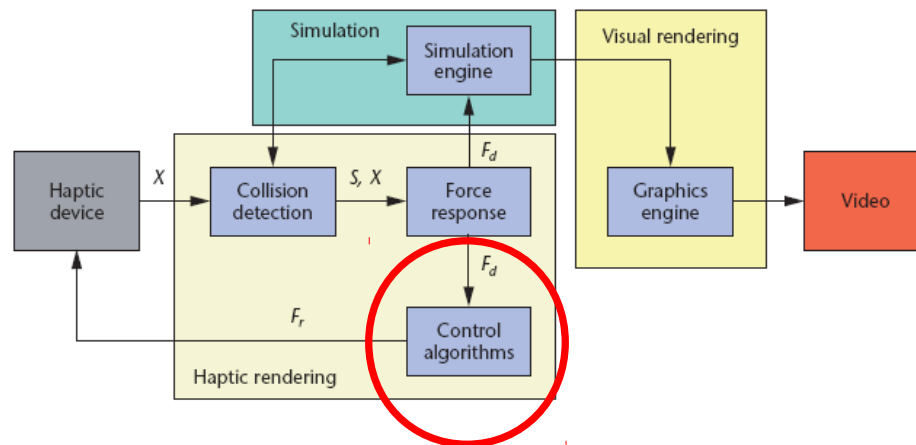


Haptic Interface Architecture

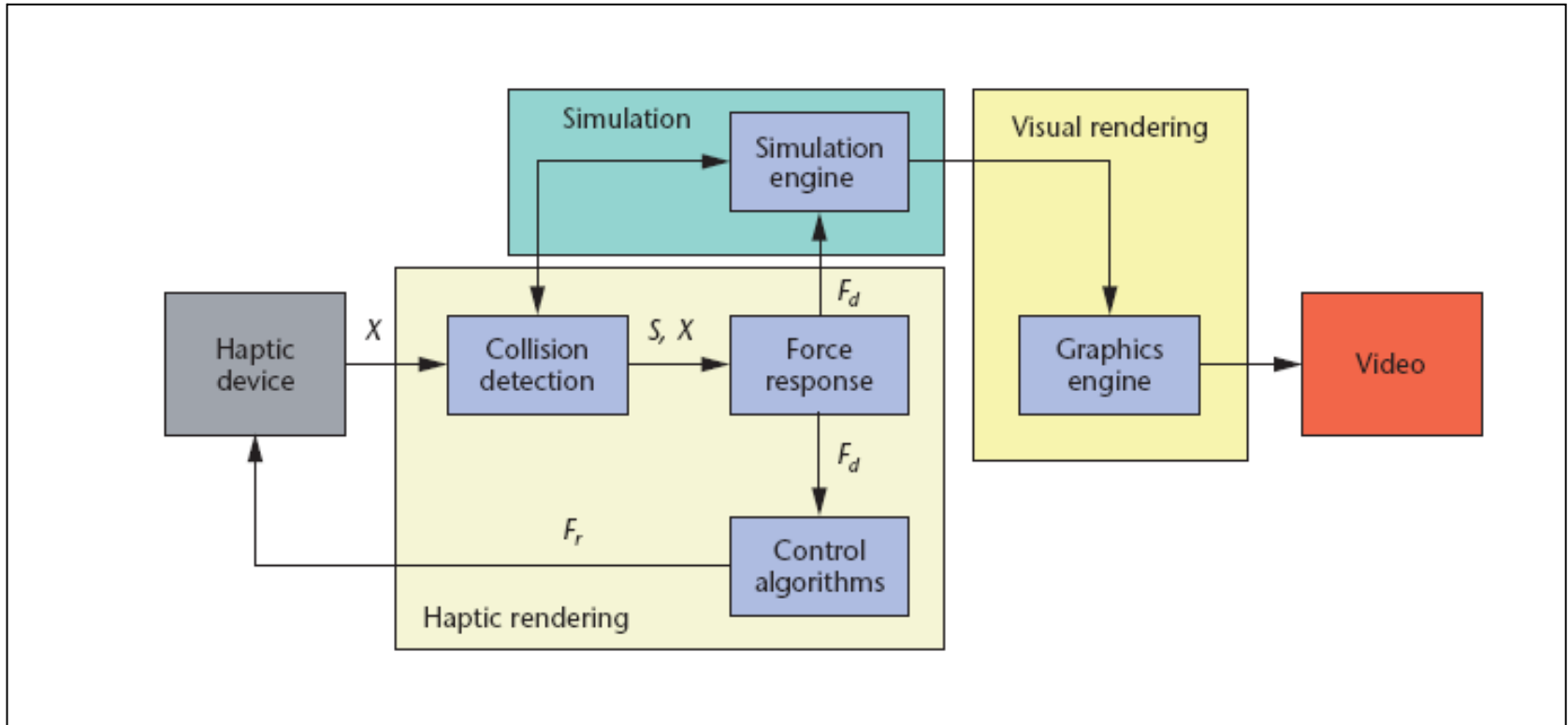
- **Control Algorithms**

- **Given a force, apply to the device so that:**

- the device remains stable
- the forces is as close as possible to what we computed



Haptic Interface Architecture

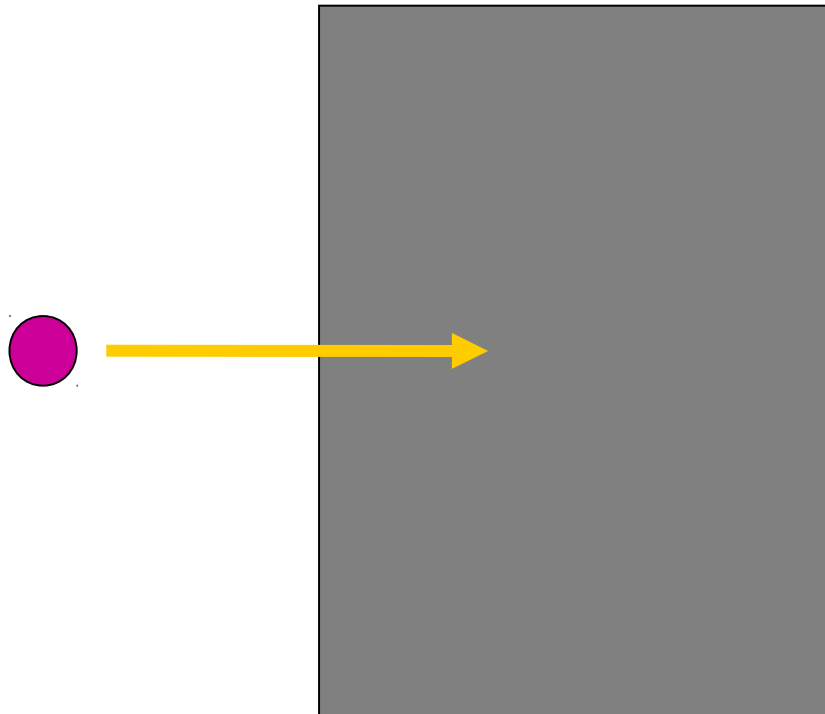


Outline

- Architecture of a Haptic Interface
- **Virtual Walls and Potential Fields**
- Issues with Potential Fields
 - Corner cases
 - Limitations of Virtual Environments
- Time is of the essence

Virtual Walls: let's discuss...

- **What's the simplest way to render a virtual plane (a.k.a. "virtual wall")?**



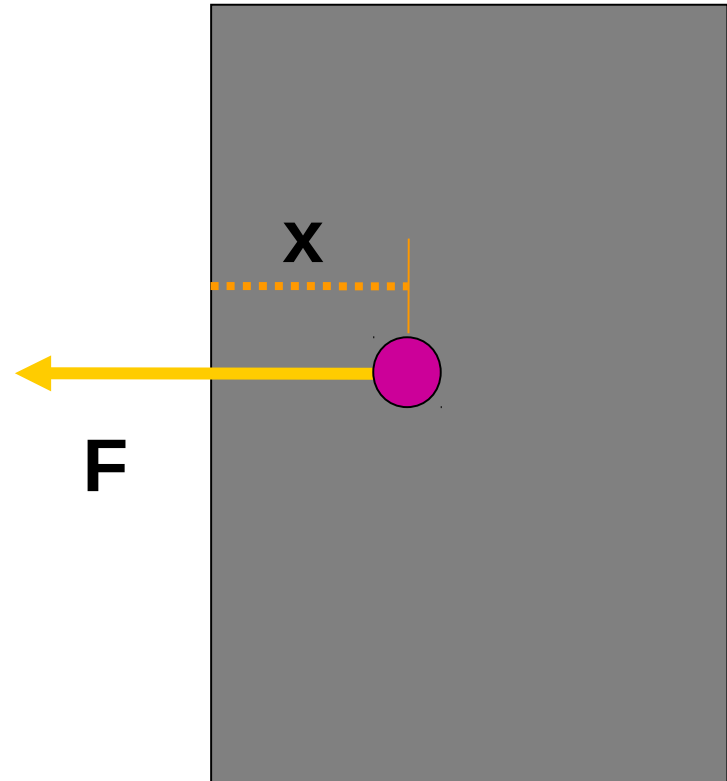
Virtual Walls

- **The simplest VE you can conceive**
 - literally a planar 3D spring
- **Useful for controls to study stability**
- **A useful building block for more complex interactions**

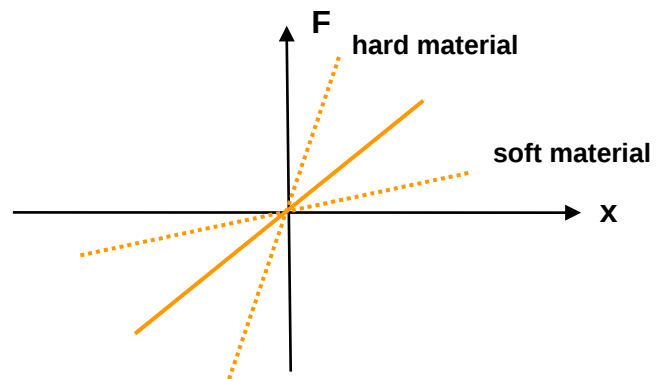
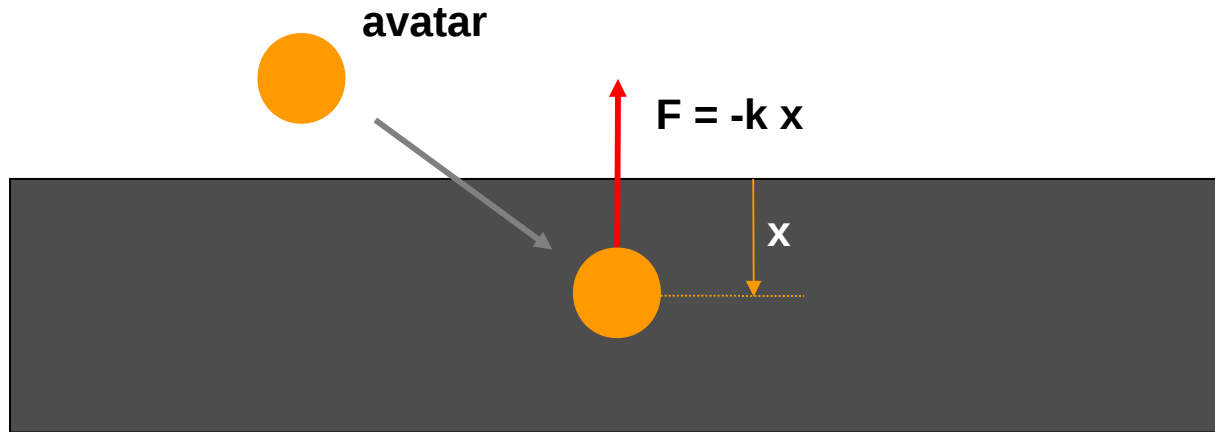
Virtual Wall Algorithm

- Virtual Wall “algorithm”

```
if ( $x < 0$ )  
     $F = -Kx$ ;  
else  
     $F = 0$ ;
```

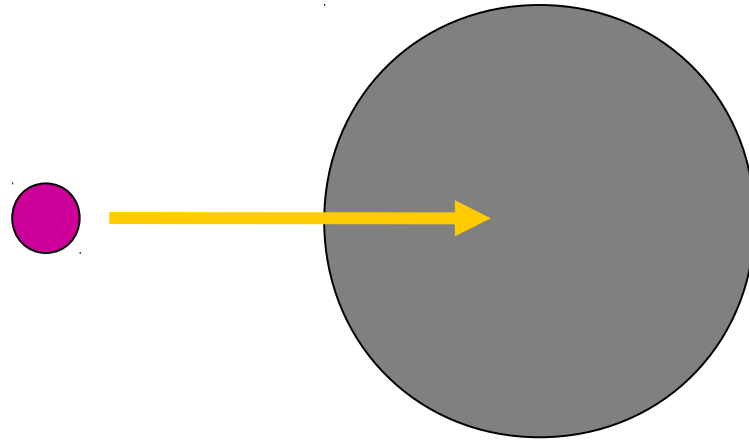


Stiffness of Virtual Walls



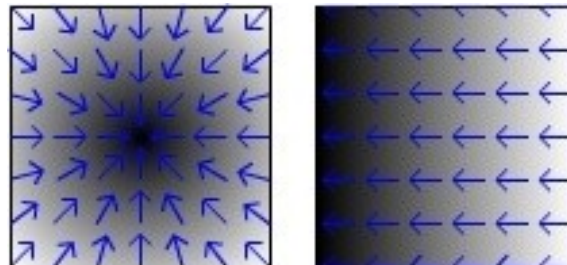
Potential Fields: let's discuss

- What's the simplest way to render a virtual cylinder coming out of the screen?



Potential Fields

- The term is “borrowed” from conservative force fields
 - a vector field representing the gradient of a potential
 - Line integral (work) of vector field between two points (A, B) is path independent
 - Line integral of $\vec{F} = \nabla V$ on a closed path is 0



Potential Fields

- **Why potential fields?**
 - they make sense intuitively (3D springs)
 - they are easy to compute
- **Examples of potential fields**
 - virtual wall is the simplest example
 - a sphere that attracts you toward its surface
 $f(x,y,z) = -K(x^2 + y^2 + z^2 - R^2)$

Outline

- Architecture of a Haptic Interface
- Virtual Walls and Potential Fields
- **Issues with Potential Fields**
 - Corner cases
 - Limitations of Virtual Environments
- Time is of the essence

Outline

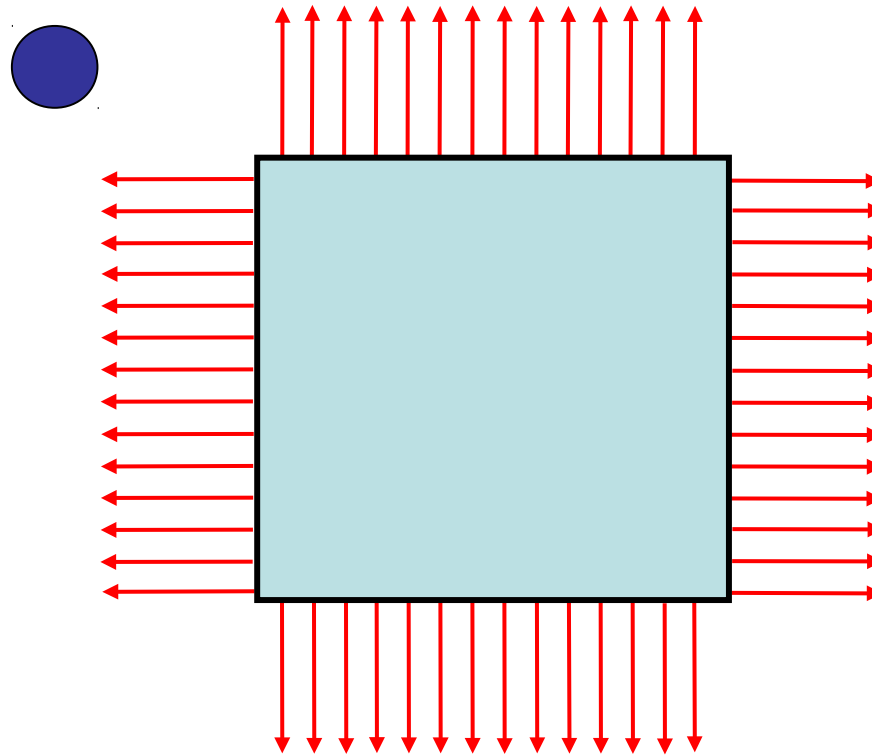
- Architecture of a Haptic Interface
- Virtual Walls and Potential Fields
- **Issues with Potential Fields**
 - **Corner cases**
 - Limitations of Virtual Environments
- Time is of the essence

Potential Fields: Issues ☹️

- so... what's wrong with potential fields?
- **Exercise together:**
 - let's think through rendering a 2D potential field together
 - draw a square

Potential Fields: Issues ☹️

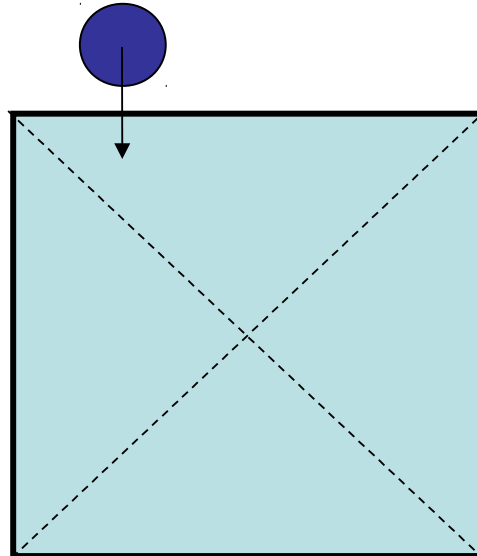
avatar



virtual object

Potential Fields: Issues ☹️

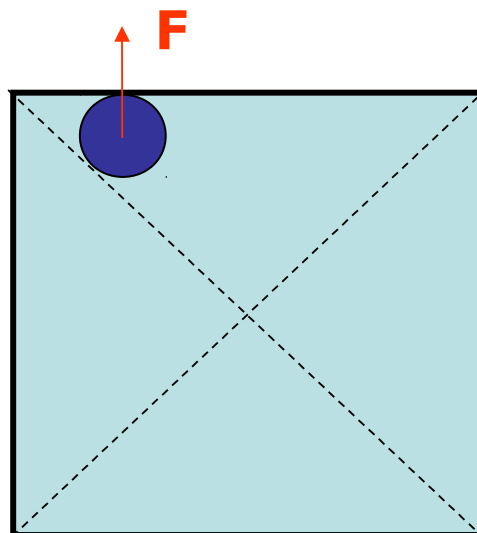
avatar



virtual object

Potential Fields: Issues ☹️

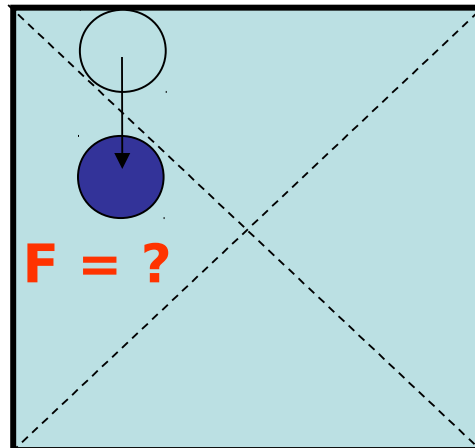
avatar



virtual object

Potential Fields: Issues ☹️

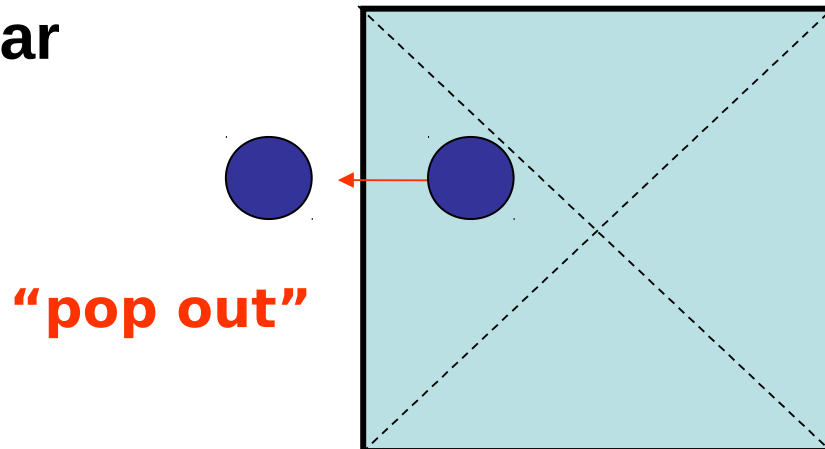
avatar



virtual object

Potential Fields: Issues ☹️

avatar



virtual object

Potential Fields: Issues ☹️

– Other Issues

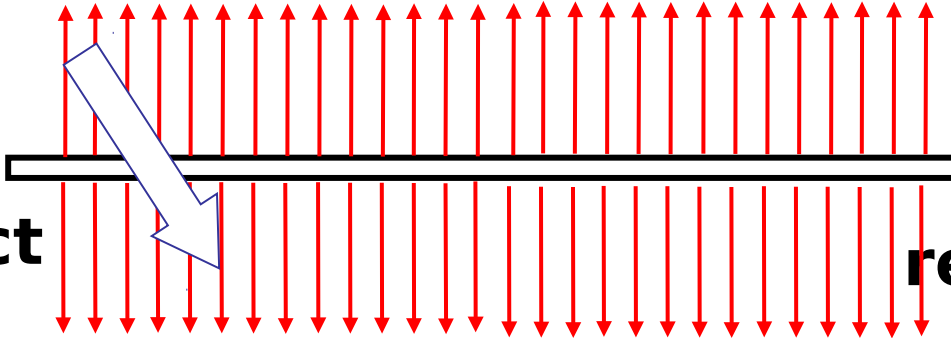
- Thin objects
- Overlapping objects

Thin Objects

avatar



“pop through” when objects are thin



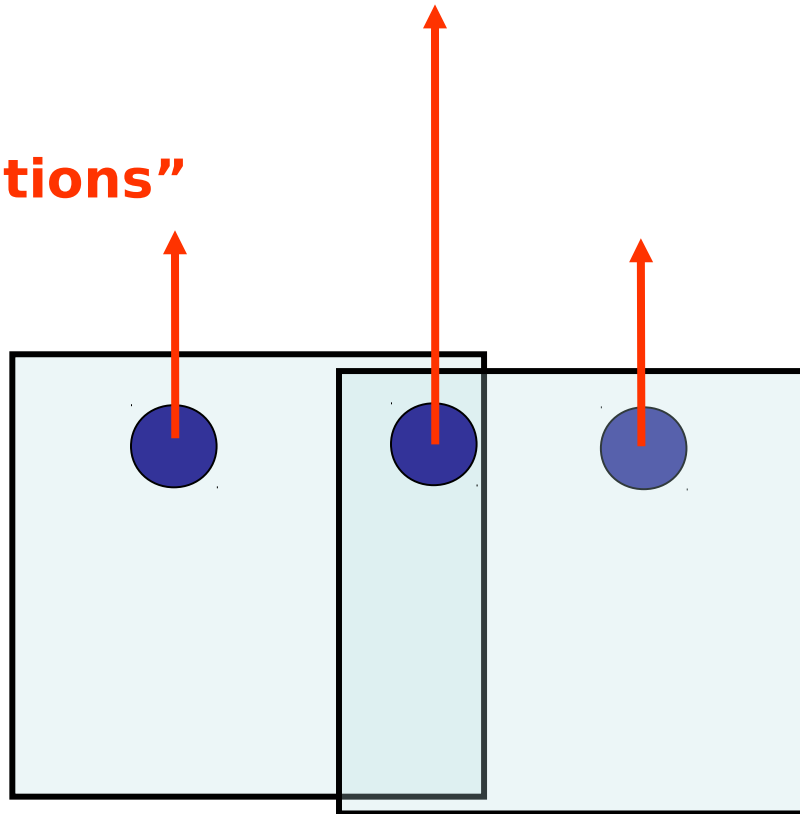
virtual object

reaction forces

Overlapping Objects

“stiffness variations”

avatar



two overlapping objects

Potential Fields: Issues ☹️

- what are the problems due to?
 - potential fields are algebraic
 - given (x,y,z) , F is fully determined
- what are we missing?
 - a concept of state, or memory
 - this will tell us “how we got into an object”
 - Penalty-based vs. Constrained-based Methods

Potential Fields: Issues ☹️

Potential Fields are an easy solution...

... with a few big drawbacks

Outline

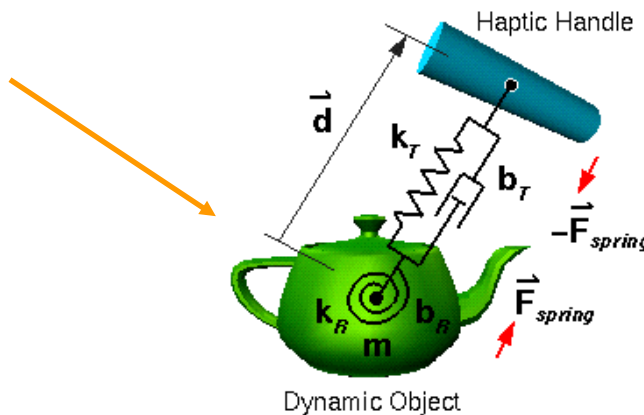
- Architecture of a Haptic Interface
- Virtual Walls and Potential Fields
- **Issues with Potential Fields**
 - Corner cases
 - **Limitations of Virtual Environments**
- Time is of the essence

More Complex VE

- How can I describe more complex VEs?

- more complex objects

- more complex contact types

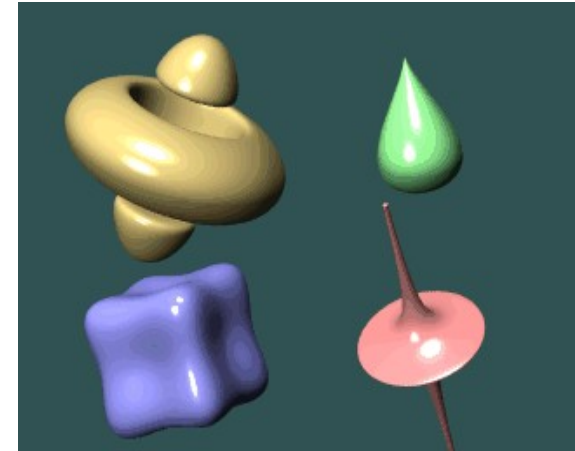


More Complex VE

- **More complex objects – typical CG formats:**
 - **implicit surfaces**
 - **triangular meshes (video games)**
 - **volumetric data (voxels)**

Implicit Surfaces

- **pros**
 - super fast collision detection
 - notion of inside/outside
- **cons**
 - visual rendering less optimized
 - harder to describe complex shapes



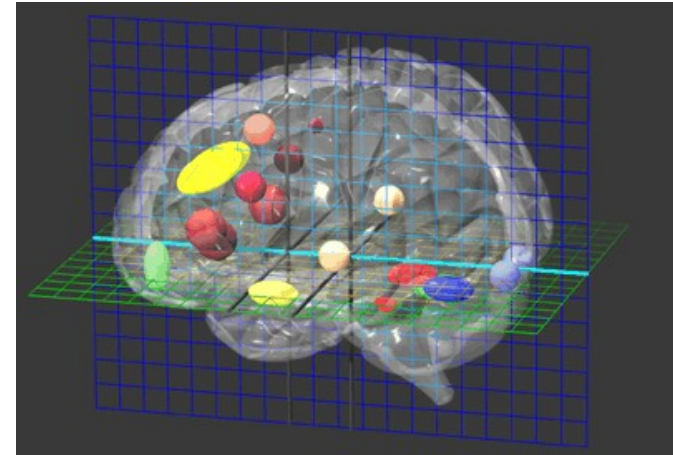
Triangular Meshes

- **pros**
 - visual rendering is super optimized
 - more availability of models and tools
- **cons**
 - no concept of inside/outside
 - collision detection is not instantaneous



Volumetric Data

- **pros**
 - fast collision detection
 - internal representation
 - concept of inside/outside
- **cons**
 - Slower visual rendering
 - Data sets are huge, not too common in consumer apps



Contact Types

- **How many degrees of freedom of force or torque can my contact apply?**
 - point contact
 - point contact with friction
 - soft finger
 - line
 - surface

Contact Types

- **Based on a point**

- **point contact:**

- 1DOF linear force
 - along the contact normal

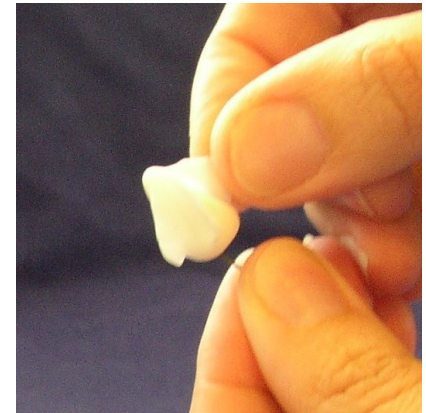


- **point contact with friction**

- 3DOF linear force
 - 1DOF along contact normal, 2 DOF in plane perpendicular to contact normal

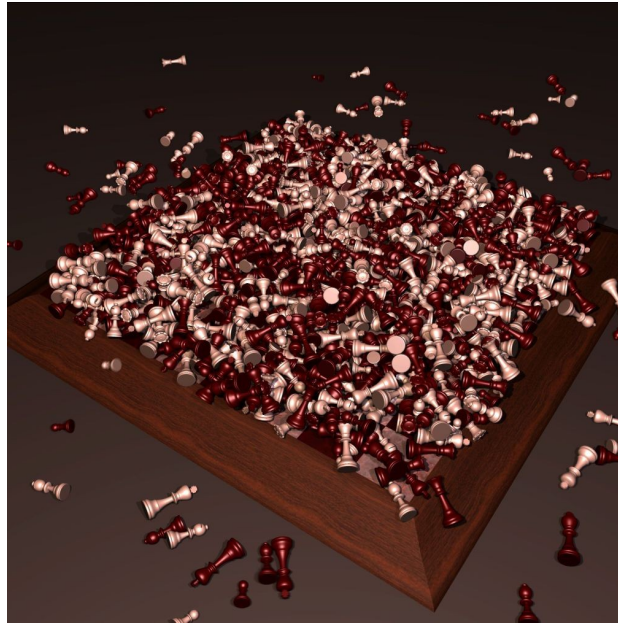
- **soft finger**

- 4DOF force (3DOF linear, 1DOF torque)
 - point contact with friction + torque about contact normal



Contact Types

- **surface**
 - 6DOF force (3DOF linear + 3DOF torques)
 - Most general rigid body contact

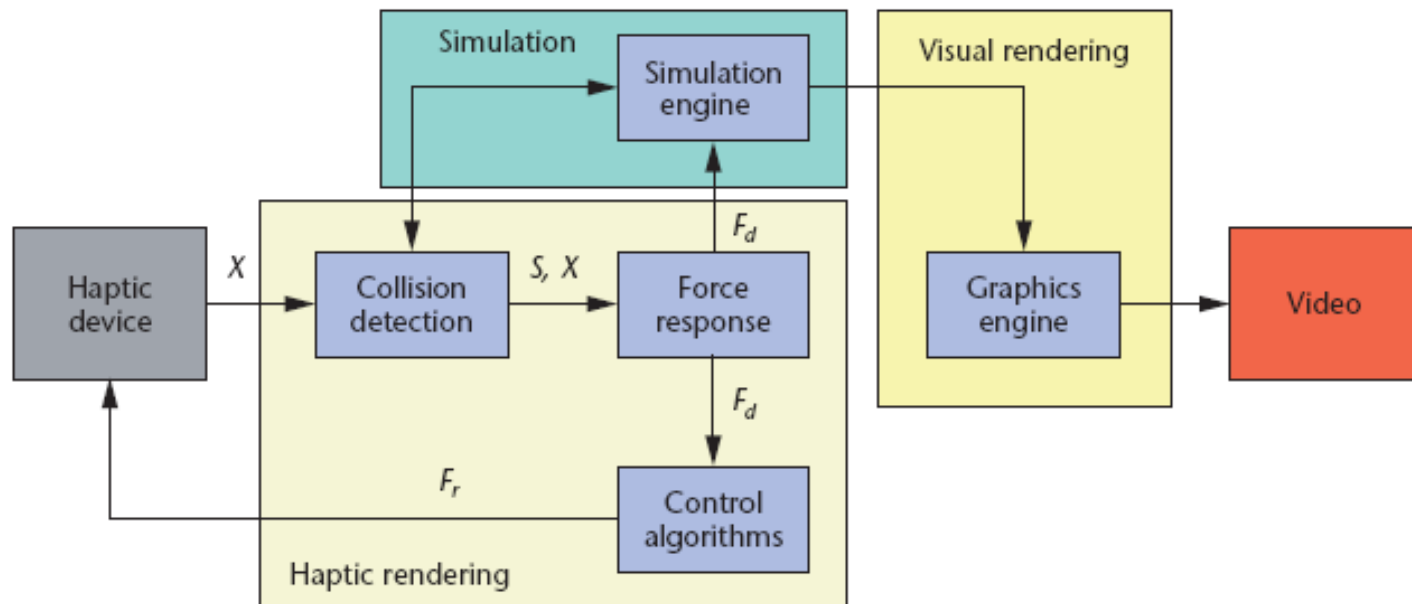


Outline

- Architecture of a Haptic Interface
- Virtual Walls and Potential Fields
- Issues with Potential Fields
 - Corner cases
 - Limitations of Virtual Environments
- **Time is of the essence**

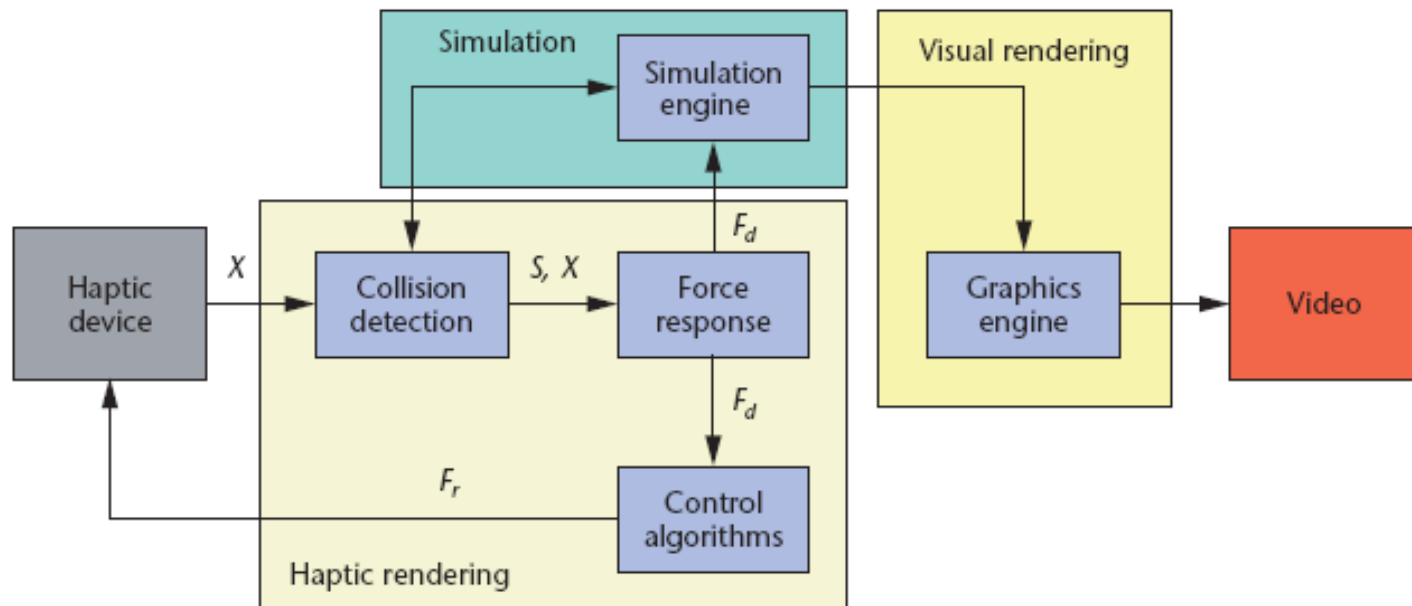
Architecture Reminder

- We render impedances
 - think 3D springs (and dampers)



Architecture Reminder

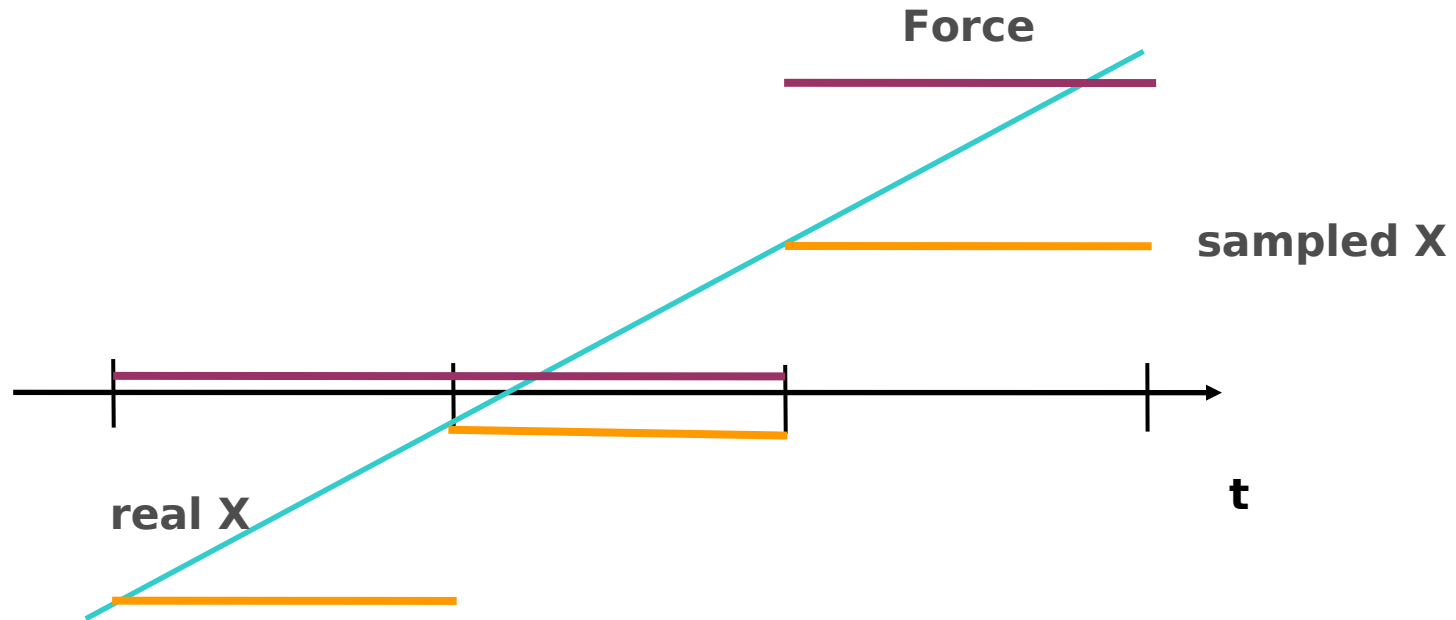
- How fast should I run my Haptic rendering loop?



as fast as possible
rule of thumb is ~1Khz

Render Springs

- What does rendering springs really mean?

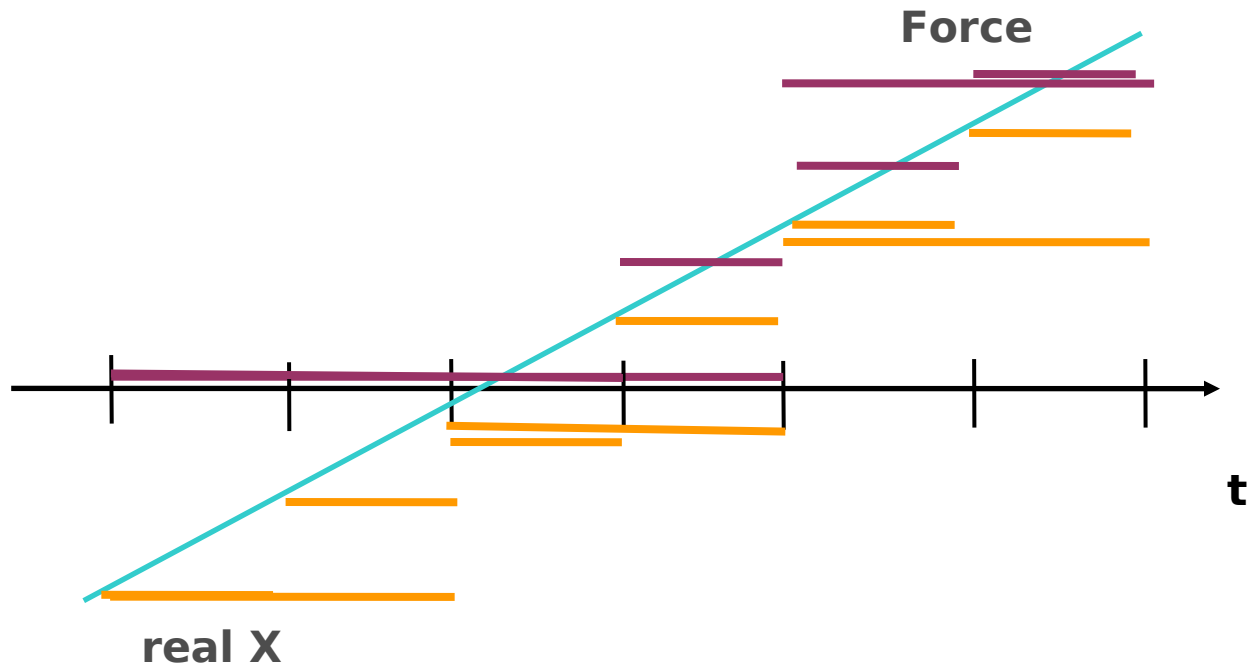


Rendering Springs

- **What does rendering springs really mean?**
- **When in contact your avatar is ALWAYS inside virtual objects**
 - the avatar is invisible
 - Fast force computation is crucial for things to feel “right”

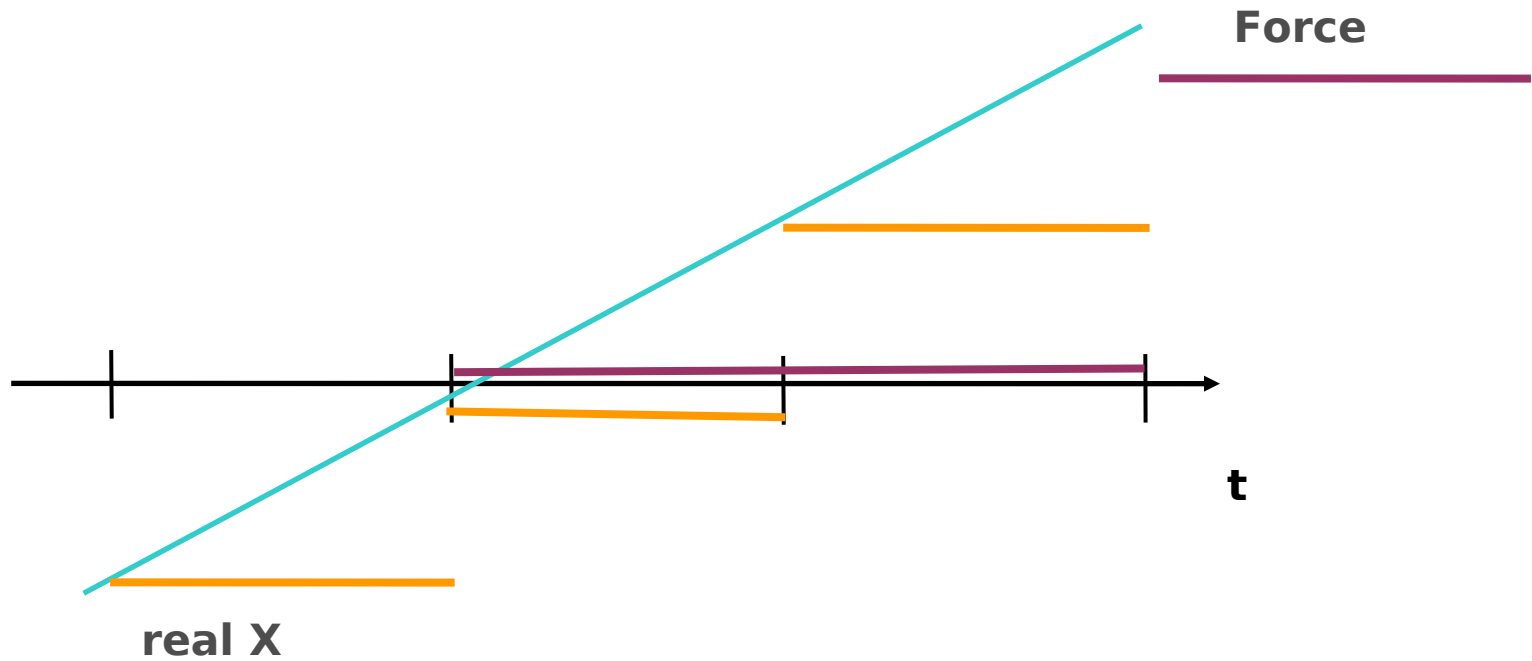
More on Speed

- Effect of servo-rate



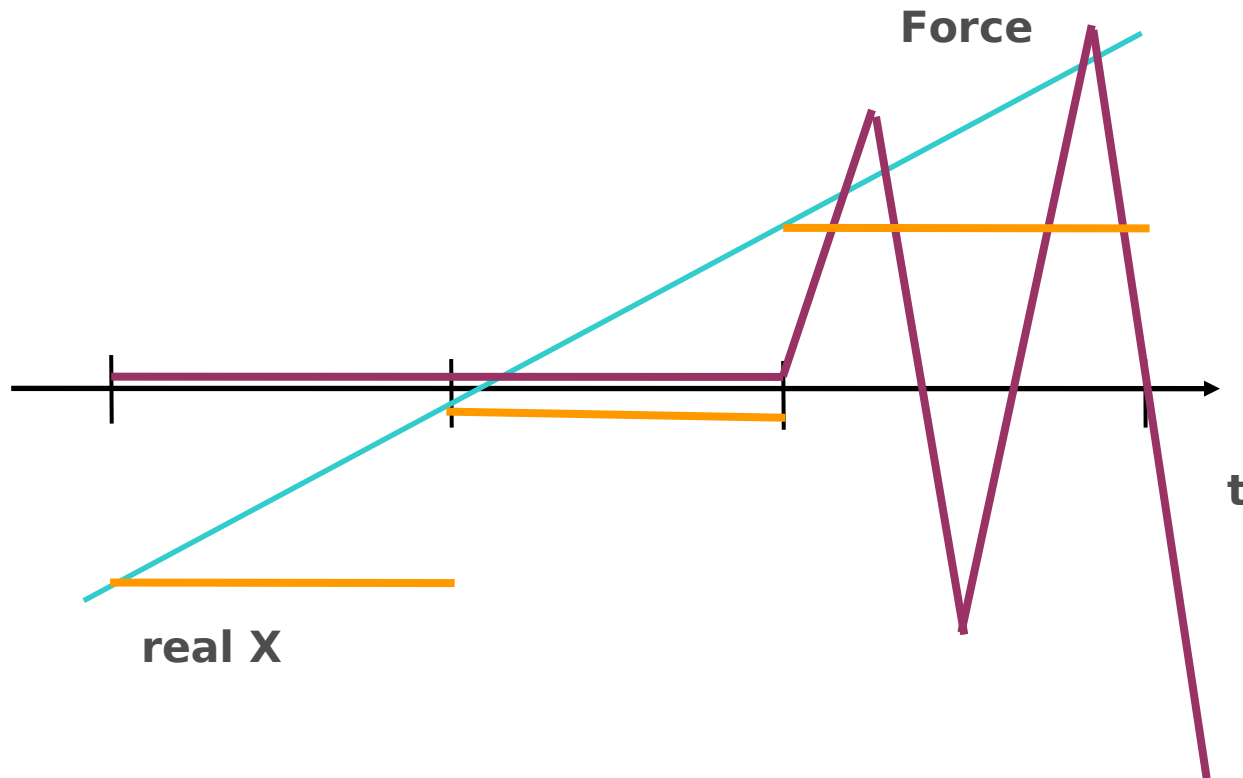
More on Speed

- Effect of delays



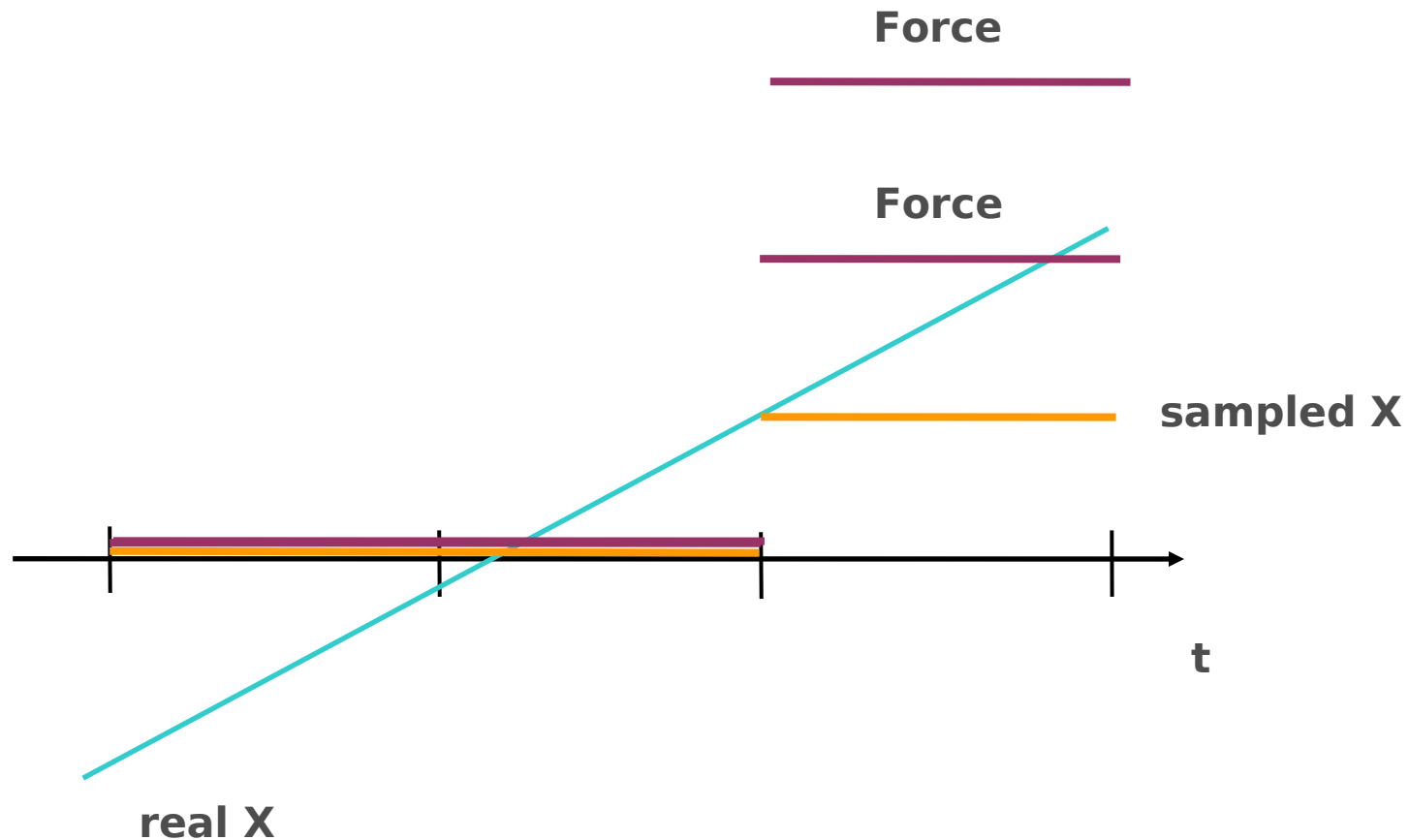
More on Speed

- Effect of servo-rate (part II)



Rendering Springs

- The higher the K the more problems



More on Speed

- How do we get speed?
 - Decoupling: 2 loops
 - Make CD and Force response as fast as possible

