# Ragdolls and IK

Eric Christensen, Insomniac Games

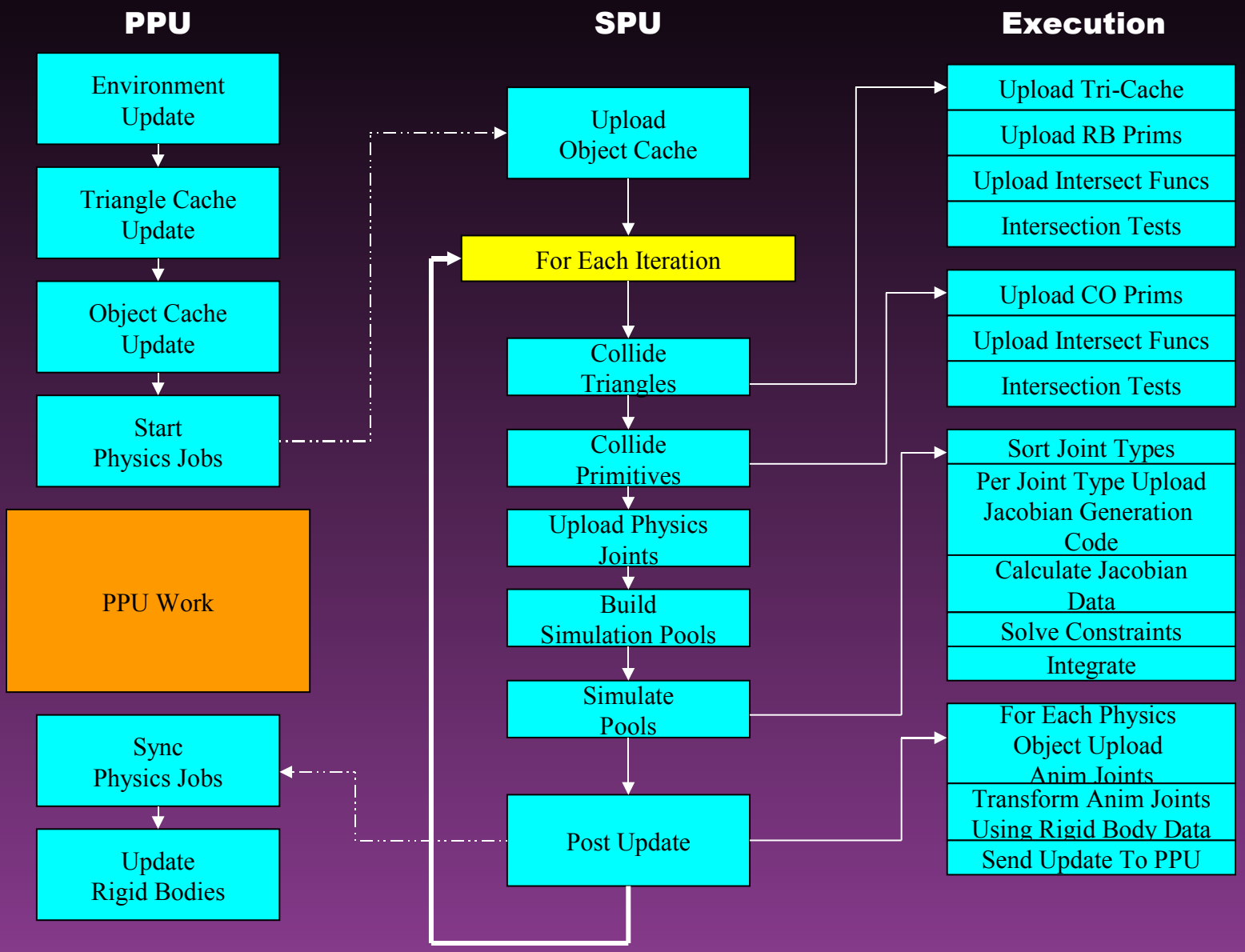
# What will be discussed

- Physics System Overview
- Joints and Limits
- Ragdolls
- IK

# Physics System Overview

- Update Pipeline
- Physics Shaders

# Update Pipeline



# Physics Shaders

- Can be written for the physics system to do specific things without having to change the physics pipeline.
- Built individually, resulting code lives in a header file that contains a byte array.
- Debug/Read-only section resides at the bottom of the byte array.
- Shaders written by gameplay programmers are registered with the physics system through a simple API.
- The physics system uploads registered shaders for a particular context at a particular stage.
- 2k of local store is allocated for the shader code, 512 bytes is allocated for the shader work buffer.

# Physics Shaders

- Shaders access a common set of functions that are passed to them, as well as pre-allocated dma tags.

  Some functions for example:

  - DMA get/put
  - Printf
  - Local store allocator
- Physics system has its own set of shaders that are native and expected in the pipeline, however, they are built the same way.
- Native physics system shaders are only slightly optimized currently and range from 2k to 12k in size.

# Joints and Limits

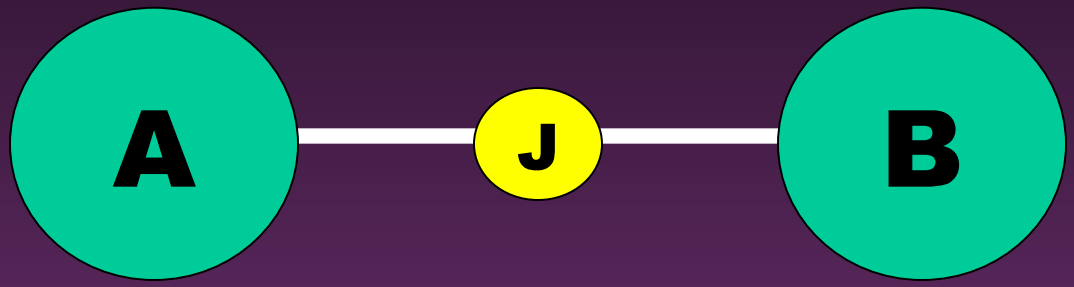
- How do Joints Work?
- Ball Joint
- Single Axis Hinge
- Dual Axis Hinge
- Lock
- Spring
- Limits

# How do joints work?

Using two rigid bodies A & B

# How do joints work?

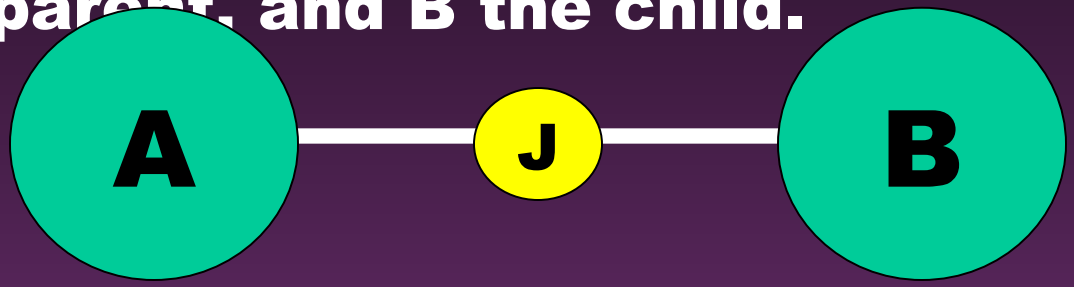
Connect them using a joint



# How do joints work?

Let's assume that rigid body A is higher in the hierarchy then rigid body B.

Rigid body A would now be considered the parent, and B the child.




One side of the joint (offset a) is local to rigid body A (parent).

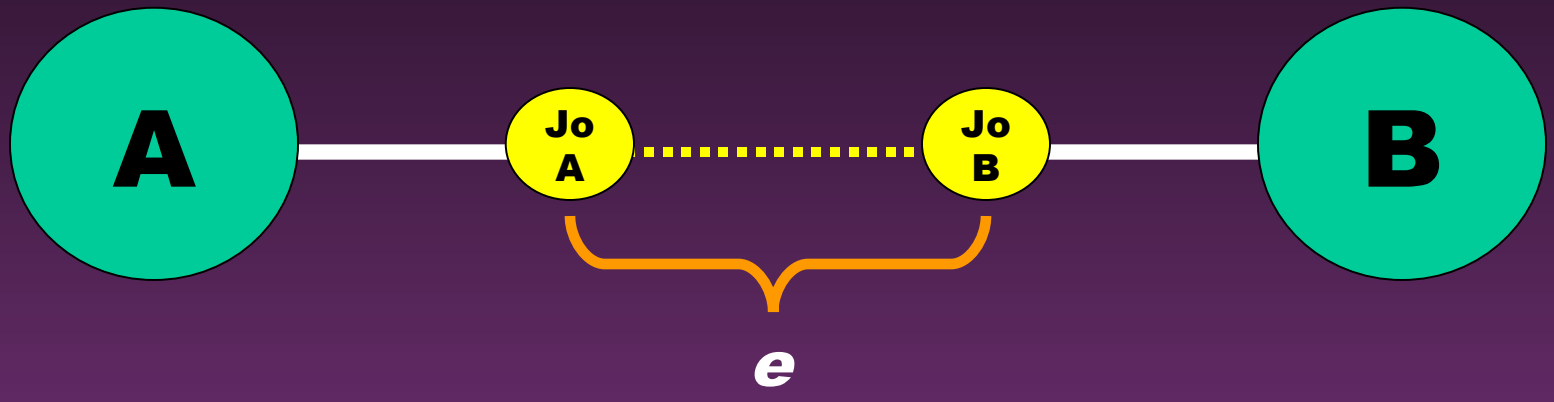
The other side (offset b) is local to rigid body B (child).

# How do joints work?

During simulation, rigid body A and B move freely, therefore, the components of the joint connecting them move with them.

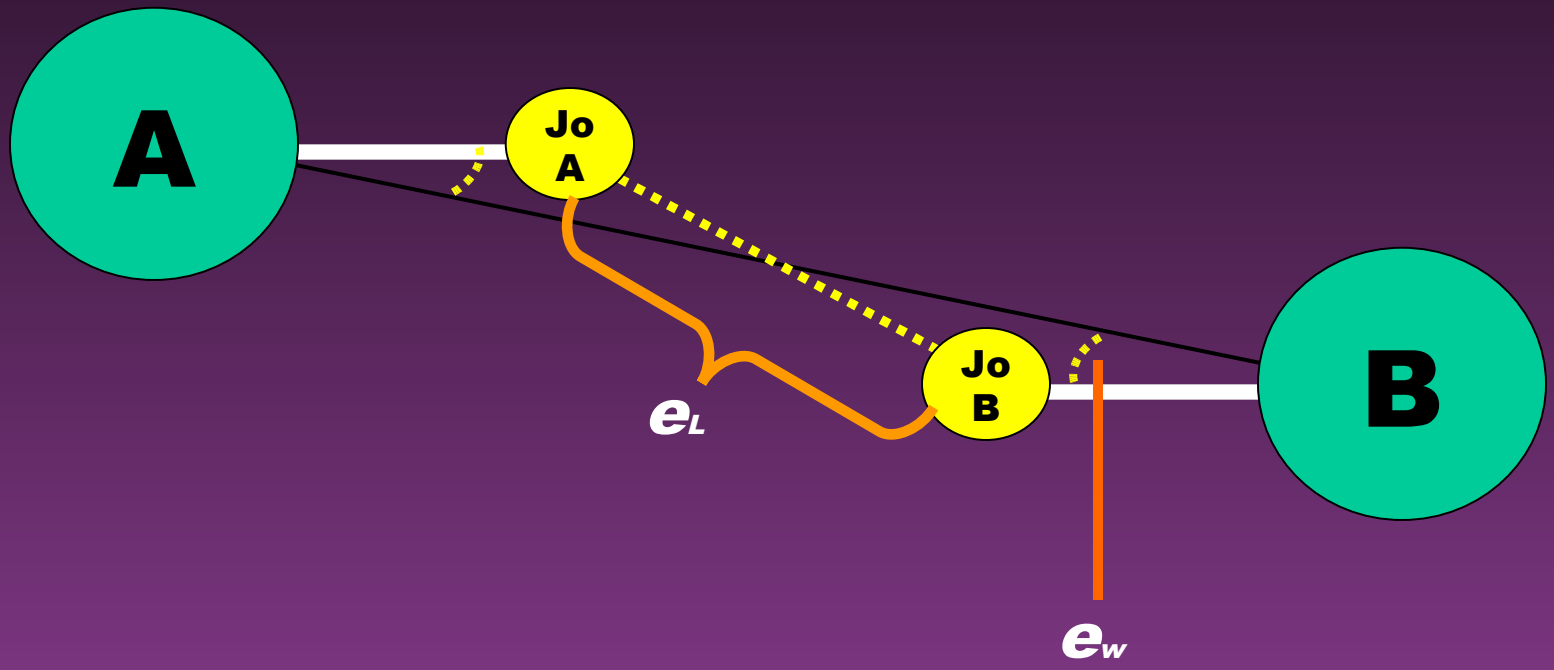
# How do joints work?

**Therefore an error vector is generated that will ideally move Joint Offset A and Joint Offset B together in world space.**



# How do joints work?

**This also takes into consideration angular error**



# Ball Joint

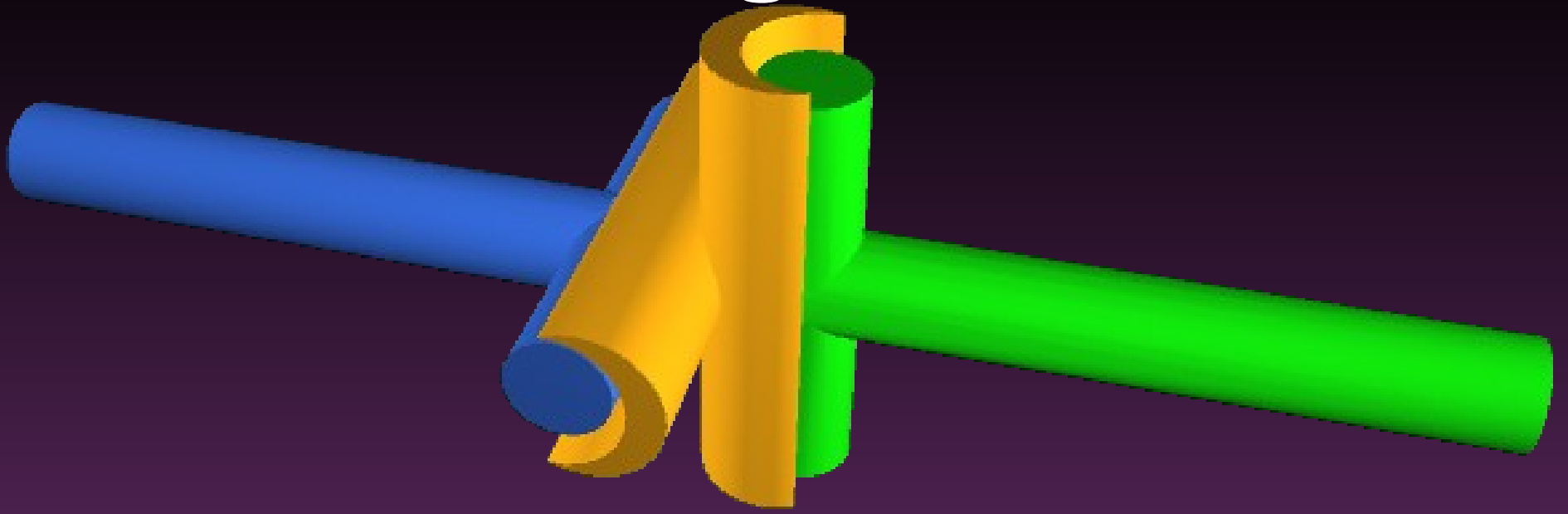
- No orientation Constraints
- Translation is Locked
- The basis of most joint types
- Swinging lamps, base of rope, etc...

# Single Axis Hinge



- Limited to rotation about a single axis
- Elbow, Knee, See-Saw, Ammo Box, etc…

# Dual Axis Hinge

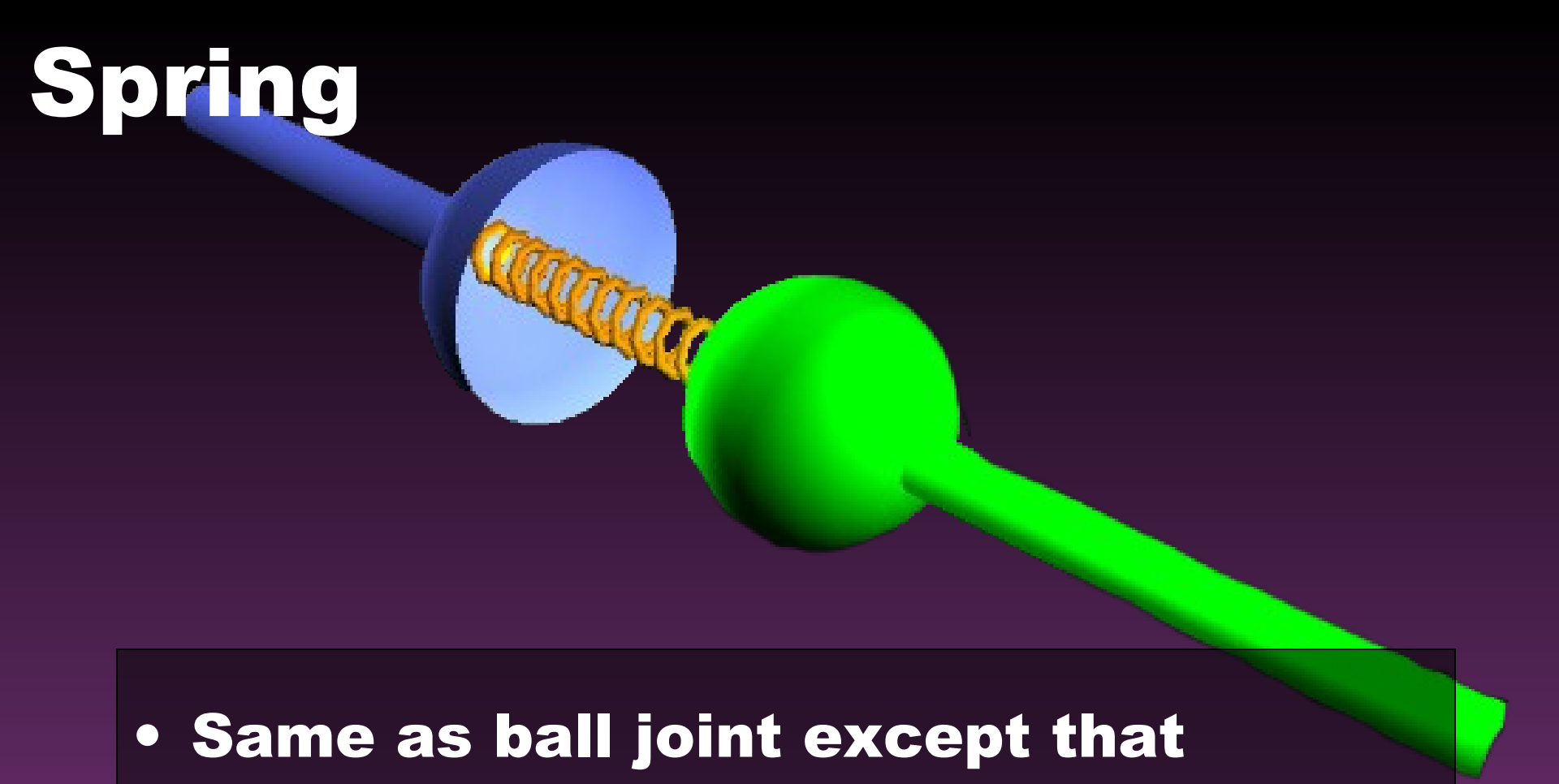


- Limited to rotation about two axis.
- Shoulder, Thigh, Chain link, Simple suspension, Really crappy merry-go-round.

# Lock

- Completely constrained orientation and translation.
- Breakable supports, IK effectors, Spikes impaling ragdolls.

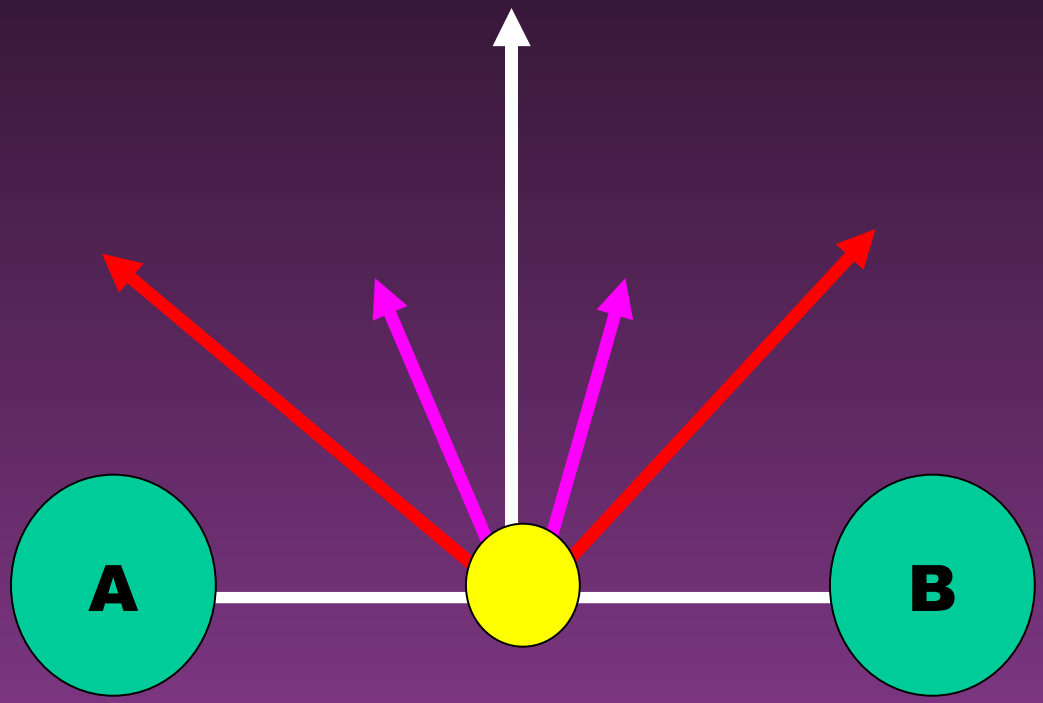
# Spring



- Same as ball joint except that translation constraint is parameterized by spring and damping coefficients.
- Stretchy rope, rubber band, etc…

# Limits

- Hard Limits
- Soft (Spring) Limits



# Hard Limits

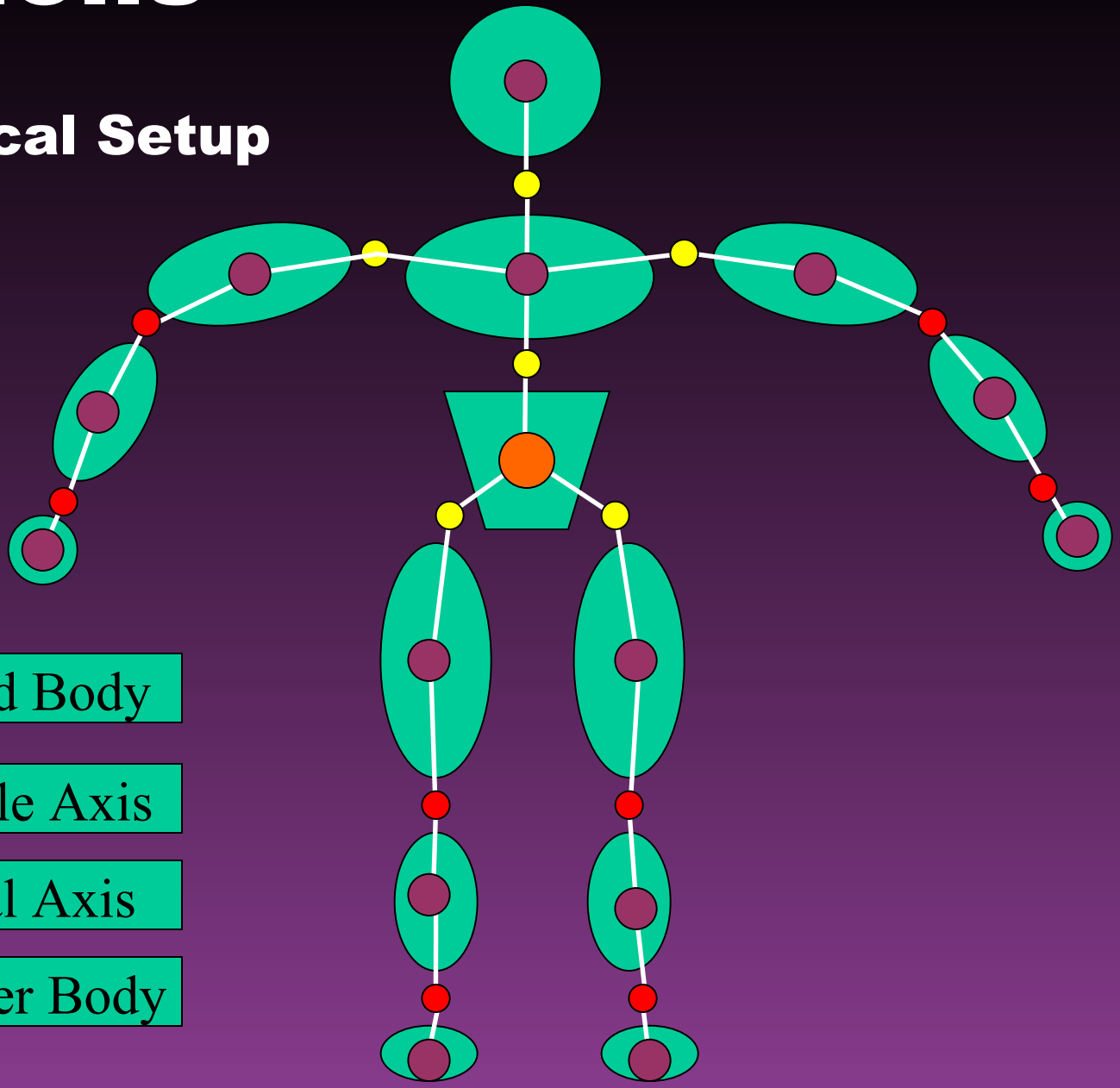
- Completely rigid angular limit.
- Attempts to solve all angular error between two rigid bodies for the limit plane.
- Extra parameters
  - Coefficient of restitution (bounce)
  - Velocity limit for restitution

# Soft (Spring) Limits

- Attempt to correct angular error between rigid bodies is tweakable.
- Default Parameters
  - Spring Coefficient
  - Damping Coefficient
- Useful For
  - Appearance of muscle tension on ragdolls
  - Vehicle suspension
  - Cabinets/Doors with spring loaded hinges

# Ragdolls

## Typical Setup



Rigid Body

Single Axis

Dual Axis

Master Body

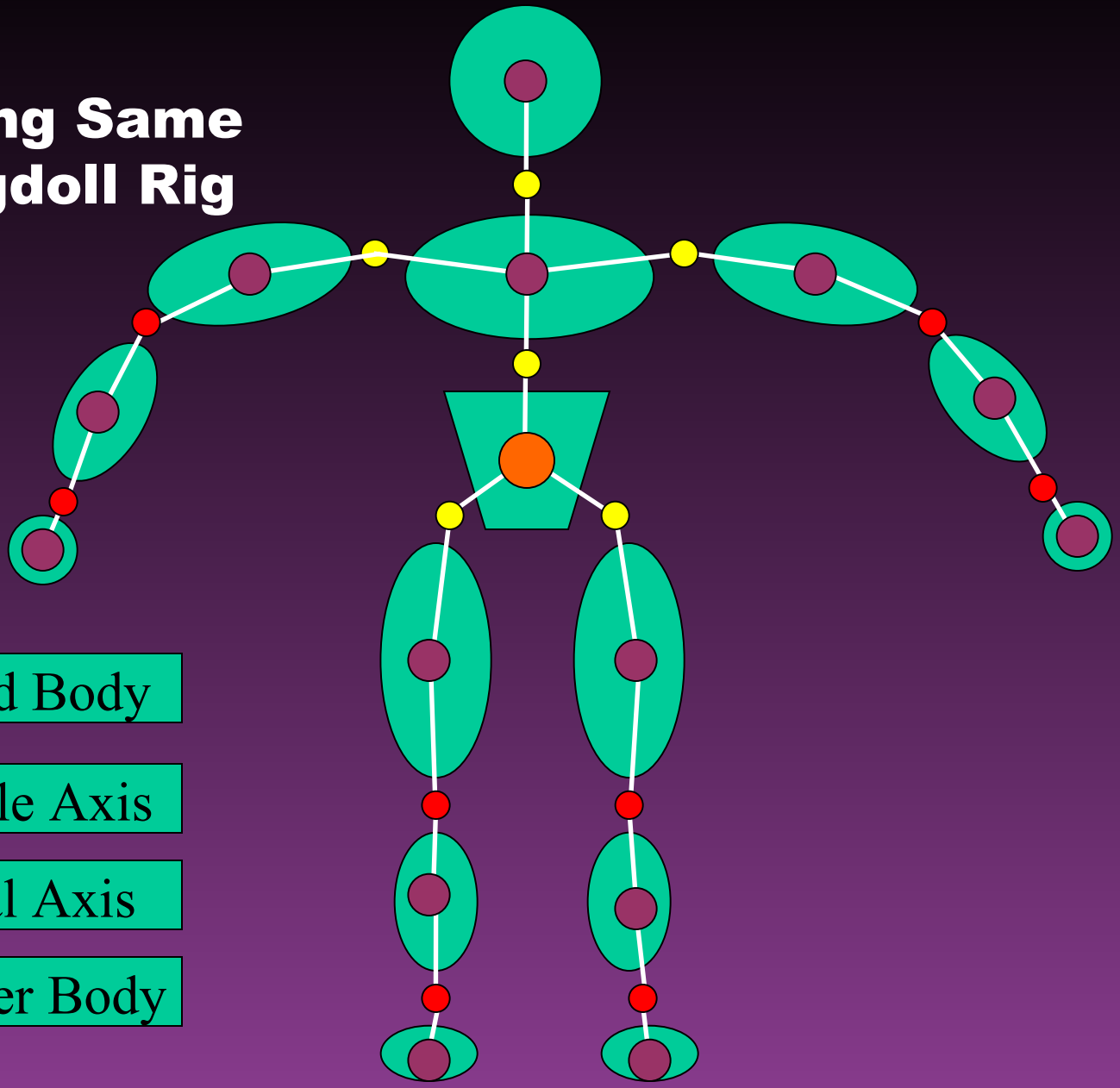
# Ragdolls

## Tips

- Try representing ragdoll with as little constraint axis as possible. This speeds up the simulation as well as adding stability.
- Try adding as little collision primitives as possible to represent a shape. This decreases the amount of constraints to solve due to generated contact points.
- For realistic limbs, set the damping coefficient on soft limits higher than the spring coefficient
- Excessive spring coefficient will result in rubbery looking limbs.
- When rigging ragdoll, first set soft limits to [-0,-0,-0] [+0,+0,+0]. This will ensure stability during simulation while you fine tune the soft limits later.

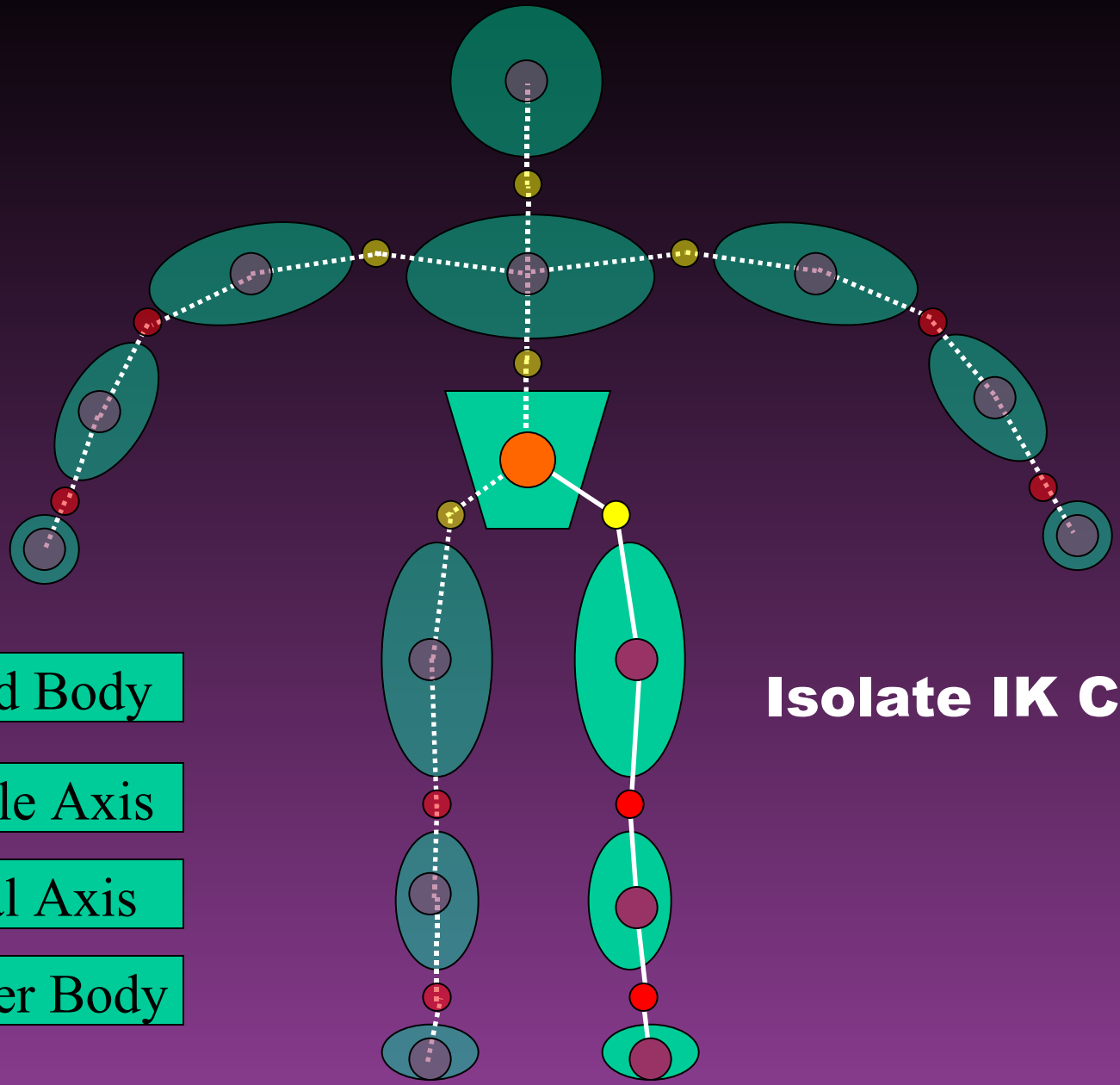
# IK

## Using Same Ragdoll Rig



- Rigid Body
- Single Axis
- Dual Axis
- Master Body

# IK

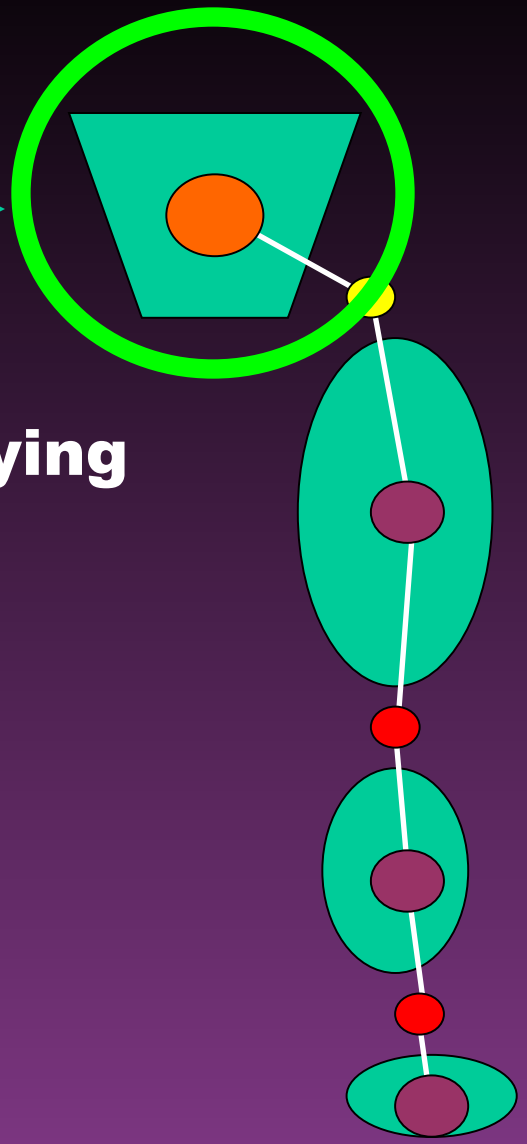


- Rigid Body
- Single Axis
- Dual Axis
- Master Body

**Isolate IK Chain**

# IK

**Root Body**



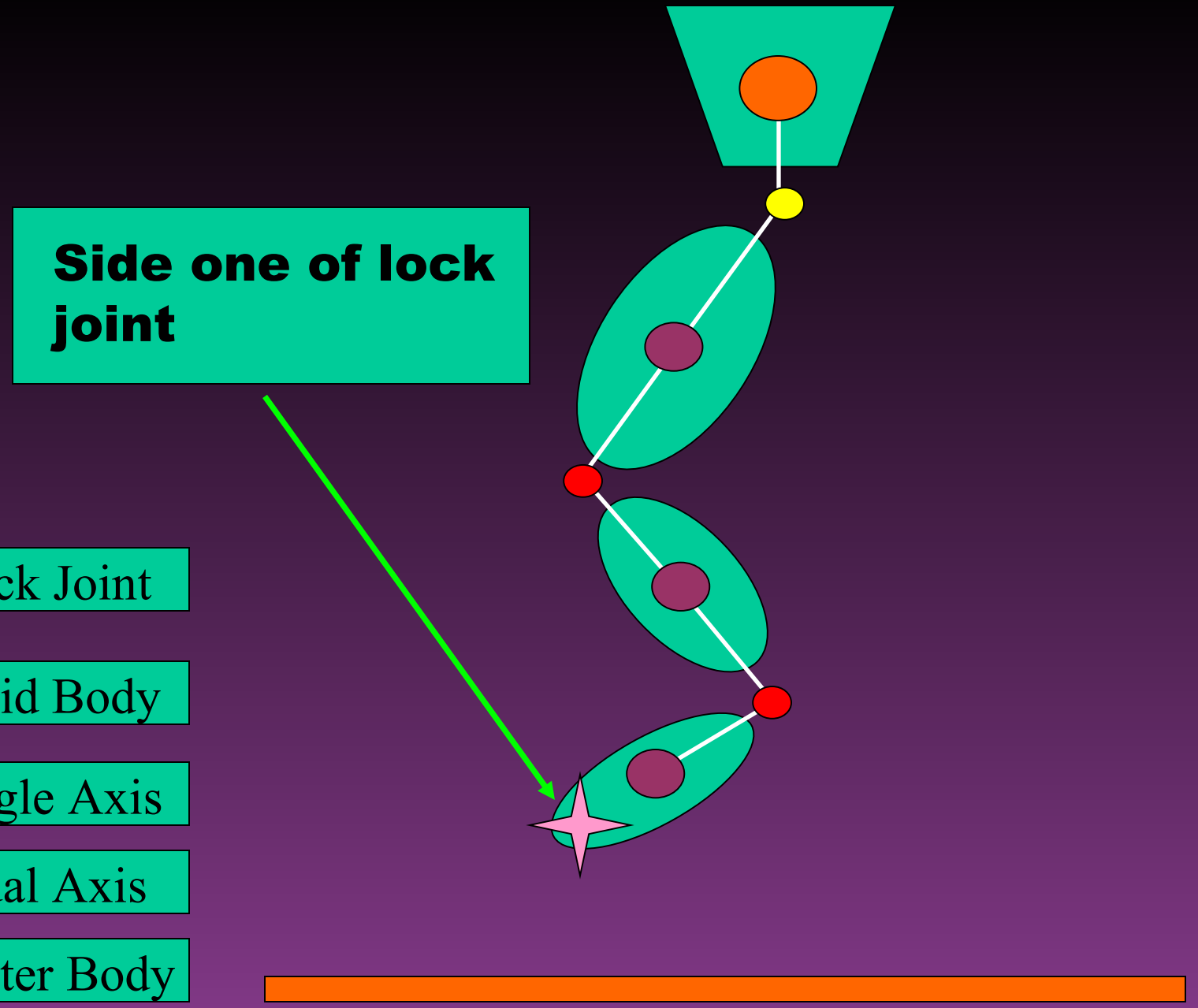
**Do This By Specifying Root Body (essentially the anchor)**

- Rigid Body
- Single Axis
- Dual Axis
- Master Body

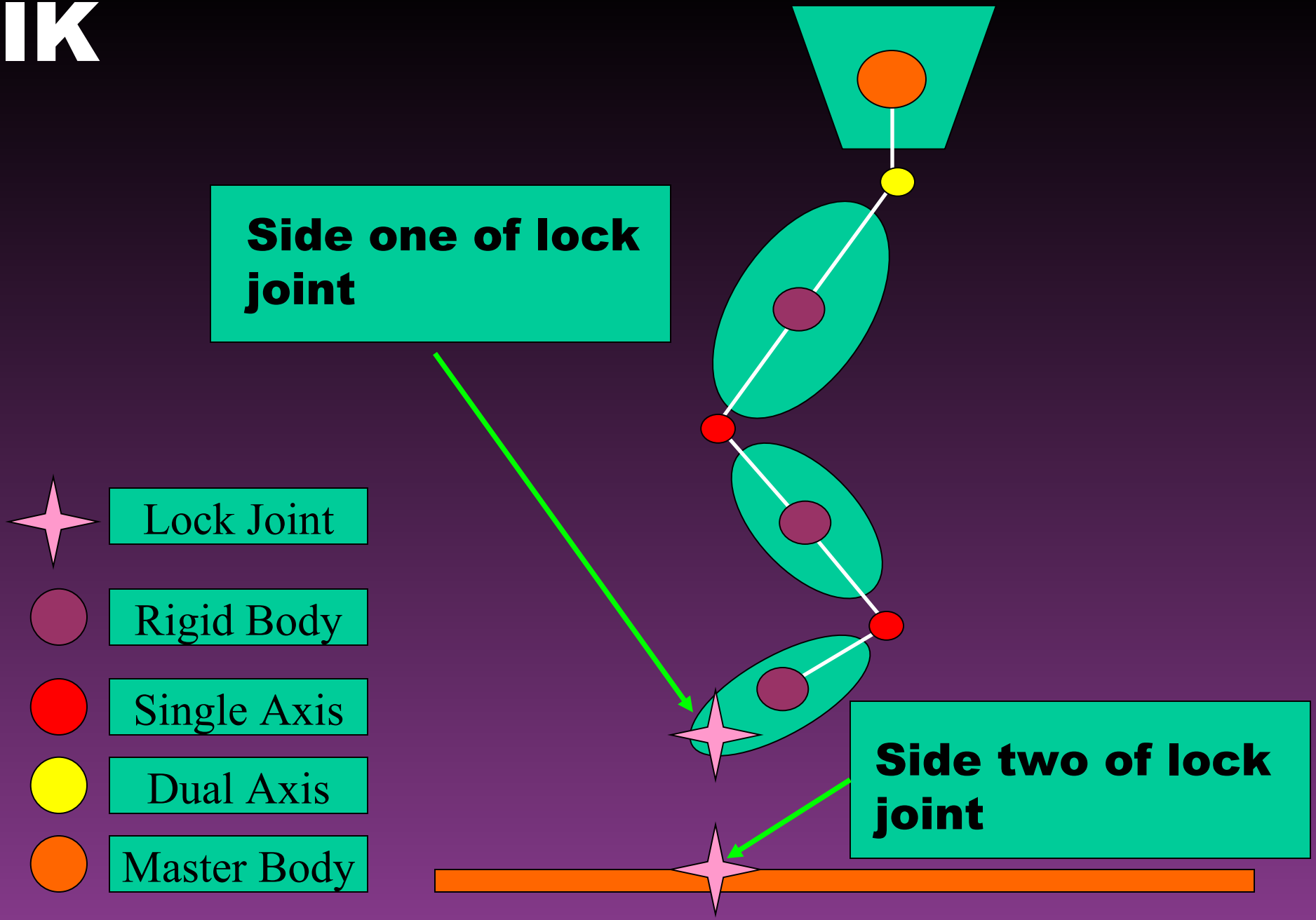
**Note:**

**Although the master rigid body is being used in this presentation, the root can be any rigid body in the ragdoll / multi-body**

# IK

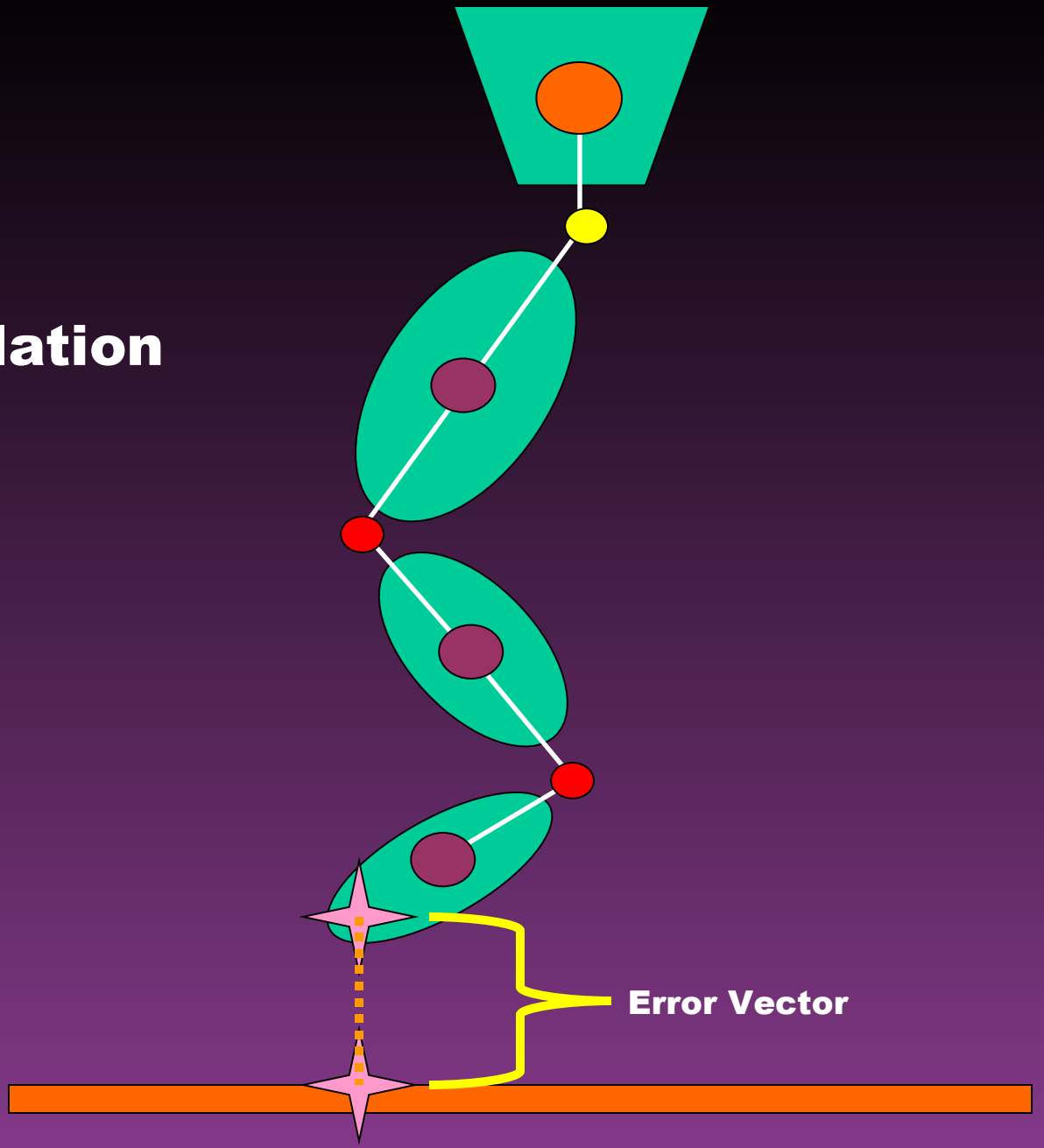

Side one of lock joint
Lock Joint
Rigid Body
Single Axis
Dual Axis
Master Body

# IK


Side one of lock joint
Lock Joint
Rigid Body
Single Axis
Dual Axis
Master Body
Side two of lock joint

# IK

## During Simulation

# IK

## During Simulation