Humanoid Control Software

Design Outline

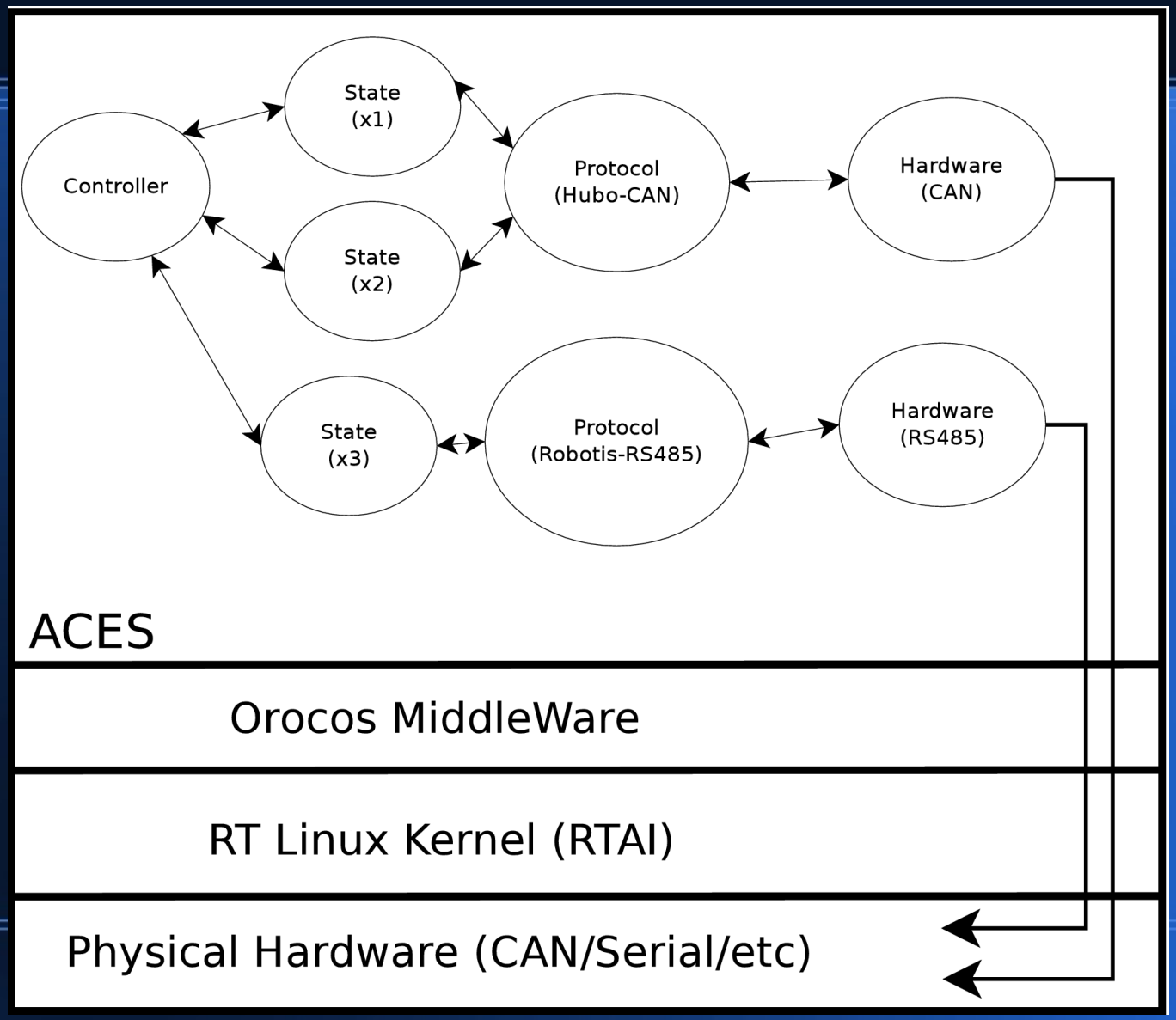
Bob Sherbert

2010 – 7 – 12

Motivation

- PIRE Grant – Humanoid architecture development
- Multiple platforms – HUBO, Miniature Humanoids, Simulators
- Desire For:
  - Experiments scaling control strategies
  - Ease of exchanging controllers
  - Fault Detection
  - Extensive logging facilities
  - Prototyping in simulation

Overview

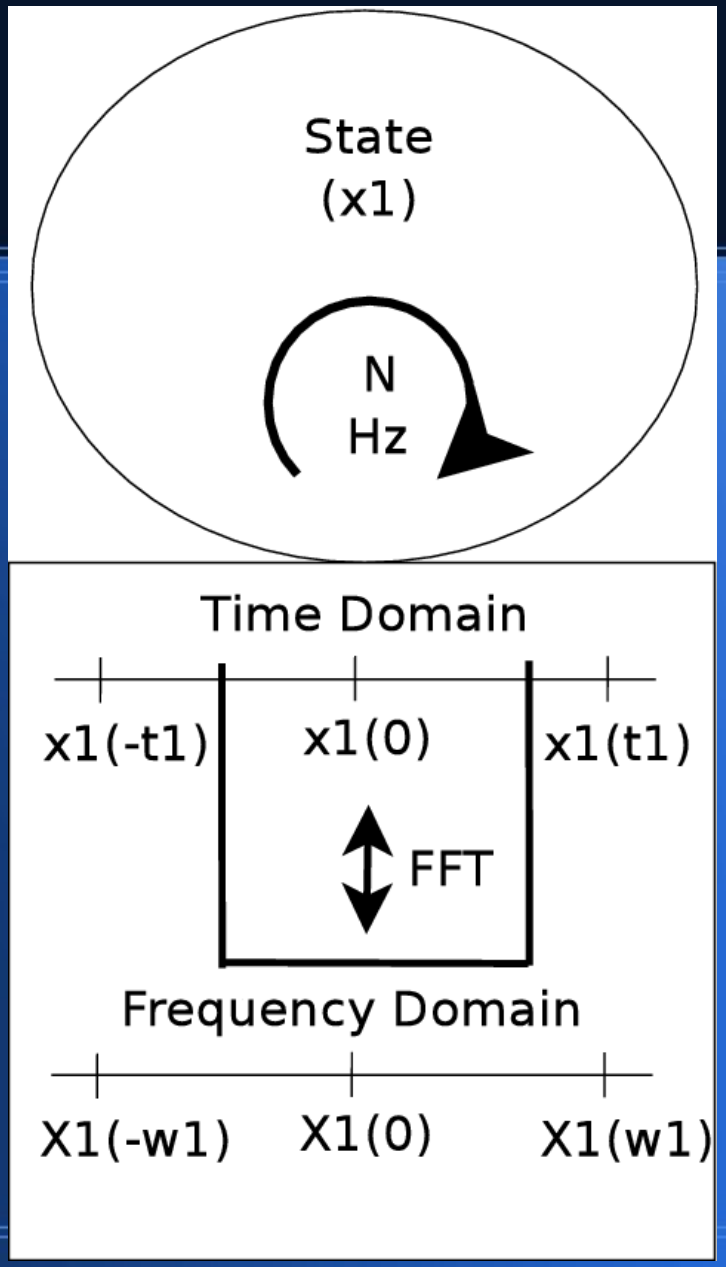

State
(x1)
Controller
Protocol
(Hubo-CAN)
Hardware
(CAN)
State
(x2)
State
(x3)
Protocol
(Robotis-RS485)
Hardware
(RS485)
ACES
Orocos MiddleWare
RT Linux Kernel (RTAI)
Physical Hardware (CAN/Serial/etc)

Overview

- Real Time (RT) services provided by the OS through Orocos
- Each instance of a component is associated w/an RT thread
- Communication between modules is event driven
  - Each component ‘subscribes’ to the events of other components
  - Information in transmissions is available to all, and each makes own decision about reception

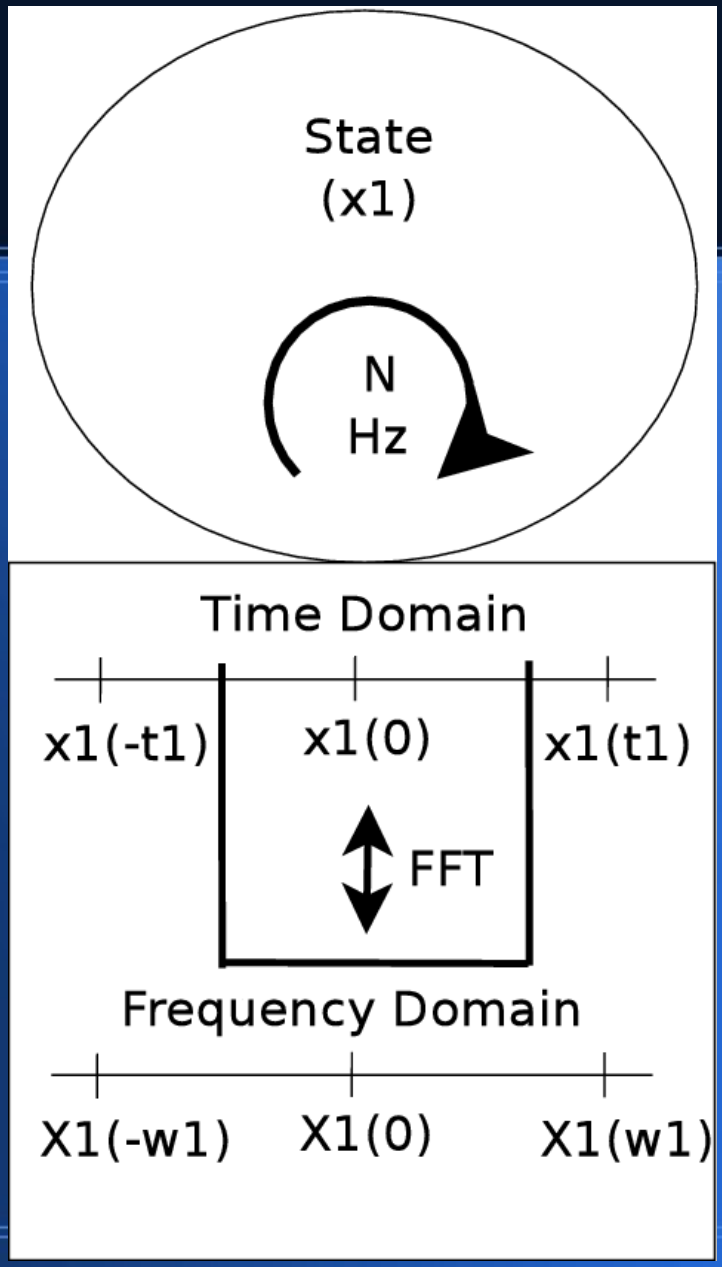
State

- Corresponds with state variable from control theory
- Presents last known value to system internals
- Independent of communication hardware (RS232, CAN, etc)



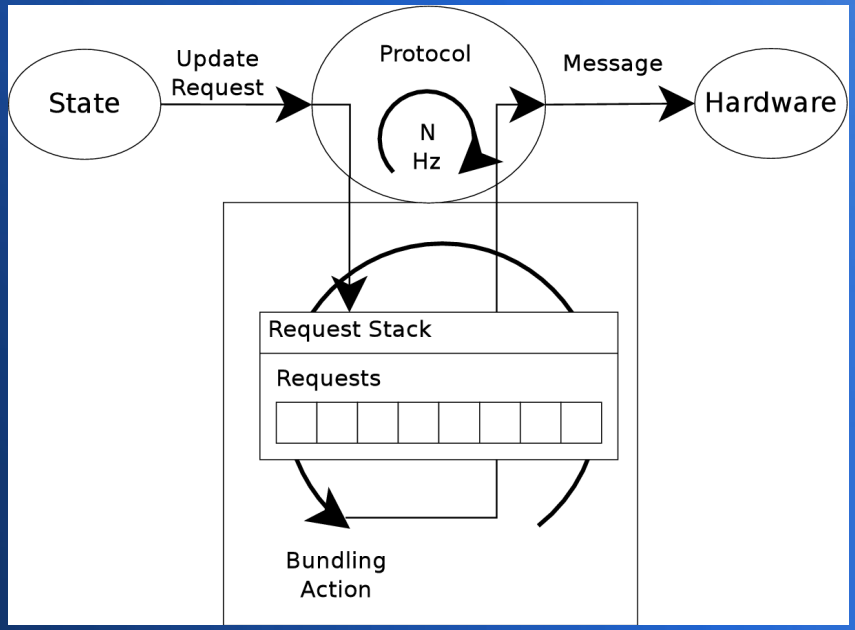
State

- Stores historical data
- Can provide averaged (smoothed) data based on historical data
- Can estimate future data based on user selected algorithm
- Compute Fourier/Laplace transforms for frequency domain controllers



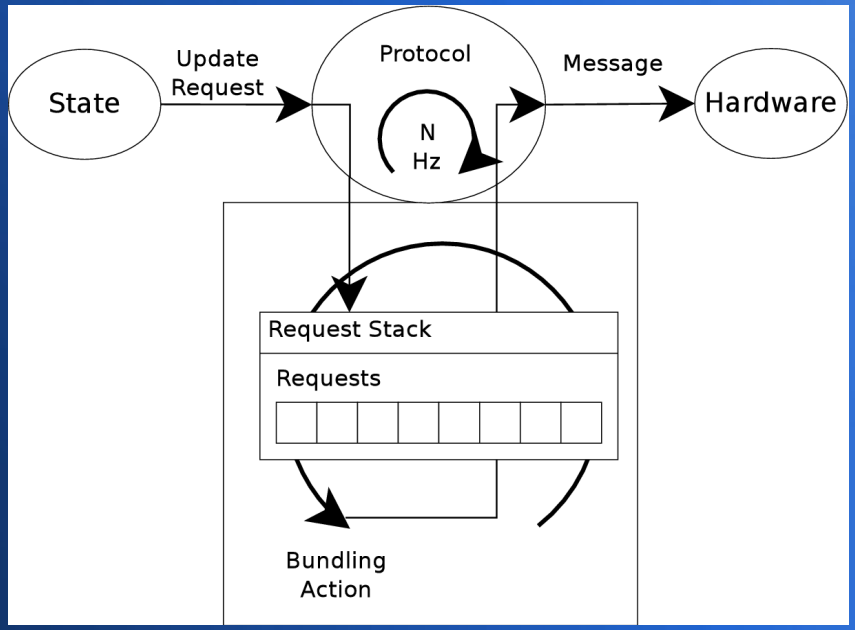
Protocol

- Describes a data format/protocol for communicating with a bus device
  - Arrangement of bytes in packet
  - Data contained within packet
- States communicate w/their corresponding devices through the protocol
  - State emits request event when update required



Protocol

- Collects, buffers, and bundles state requests
- Converts bundles into Messages that the Hardware can transmit to physical devices
- Emits Message as event – Hardware picks up event

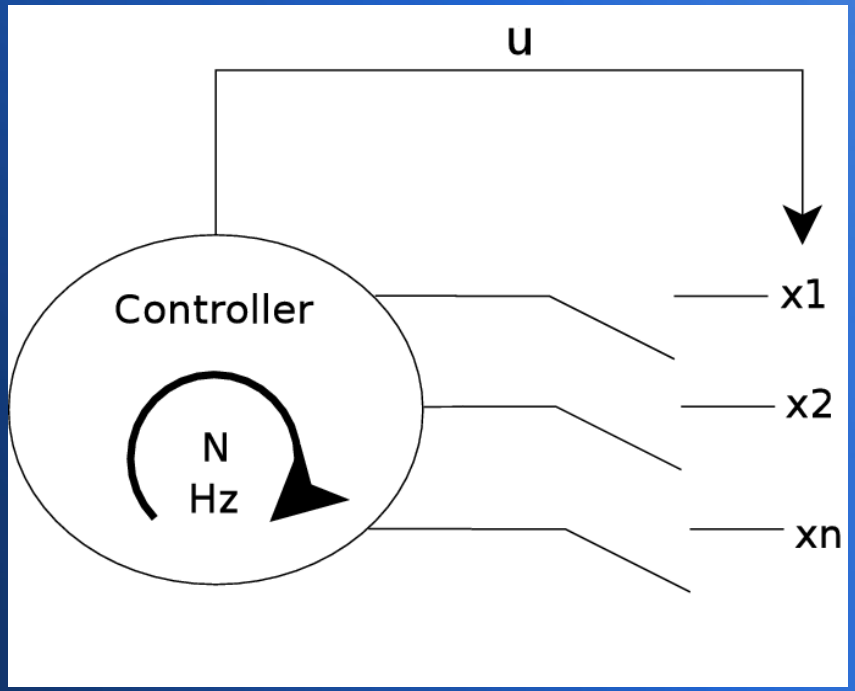


Hardware

- Gatekeeper to physical hardware
  - Prevents multiple access to line in the thread-based environment
- Converts messages from protocol into packets on the physical bus
- Tracks expected responses
- Handles Tx/Rx error reporting to higher levels

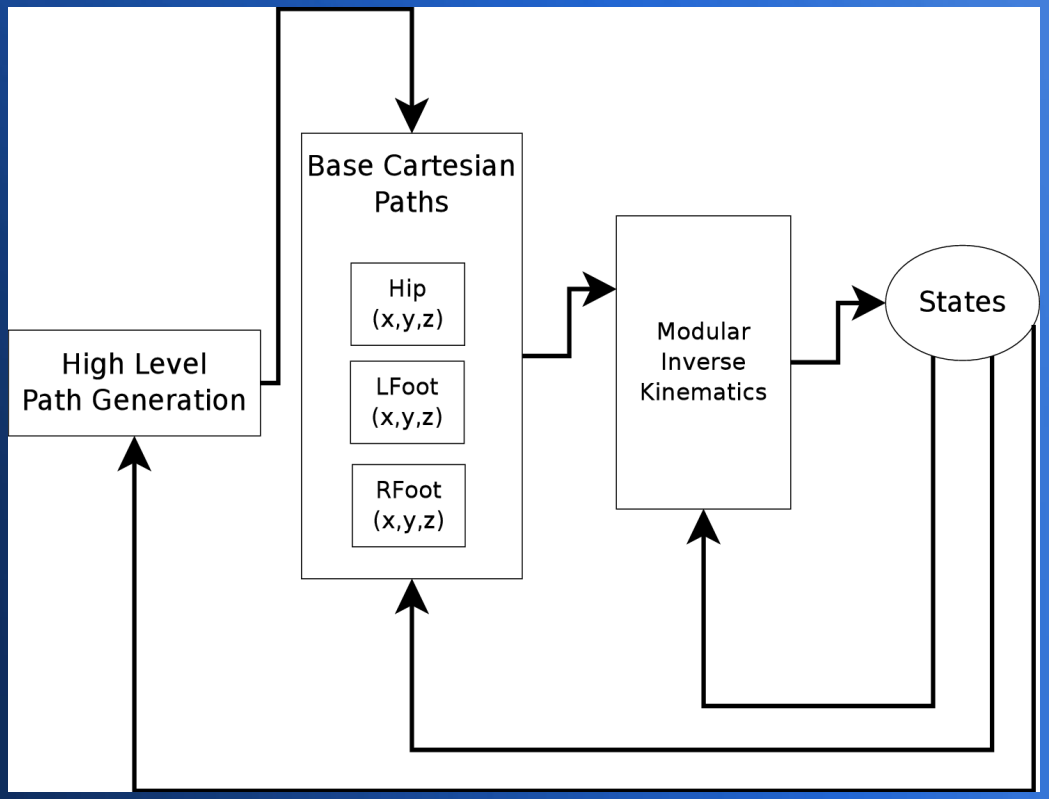
Controller

- Controller implements the feedback laws
- No restrictions placed on controller internals
- May draw from states, external data sources, algorithms, etc
- The determined control signal is broadcast to the entire system
- Subscribers take appropriate action to realize the control signal



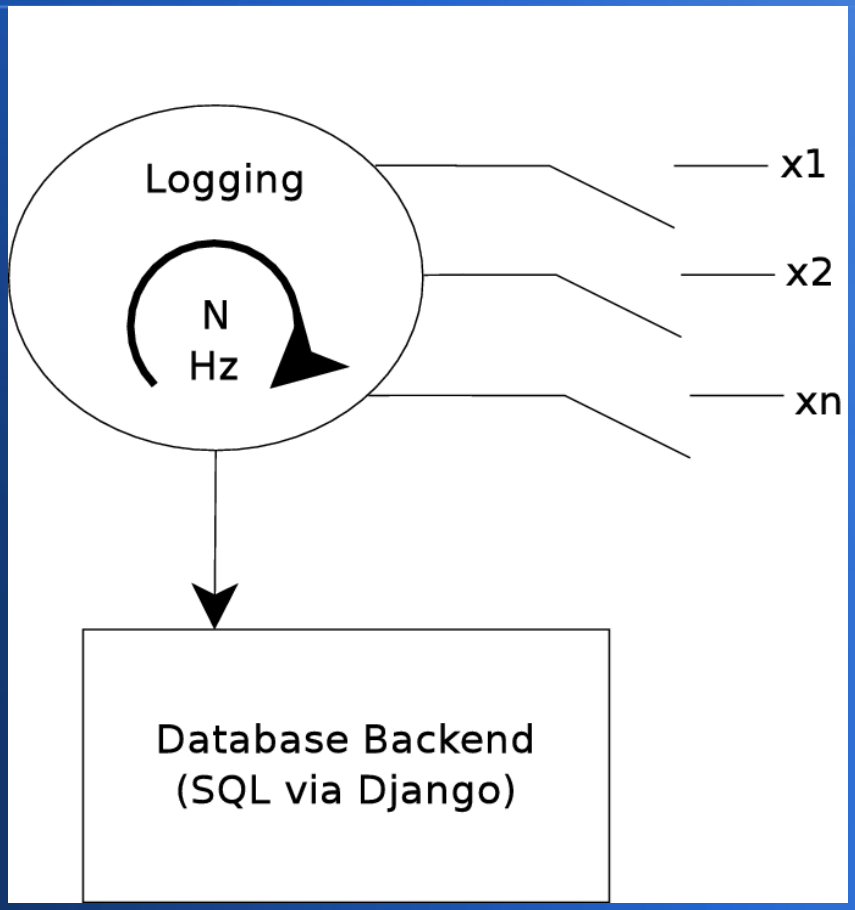
Template Controller (ex.)

- Modular design for humanoid control
- Cartesian path, IK, and joint space are each modules
  - Different module used for Hubo/Mini/Virtual
- Feedback can be provided at any level
  - Landing control vs Ankle roll control



Logging

- Internal sampling of states
- Data stored to an SQL database
- Data easily accessible through web
- Easily exported to analysis packages (Matlab)

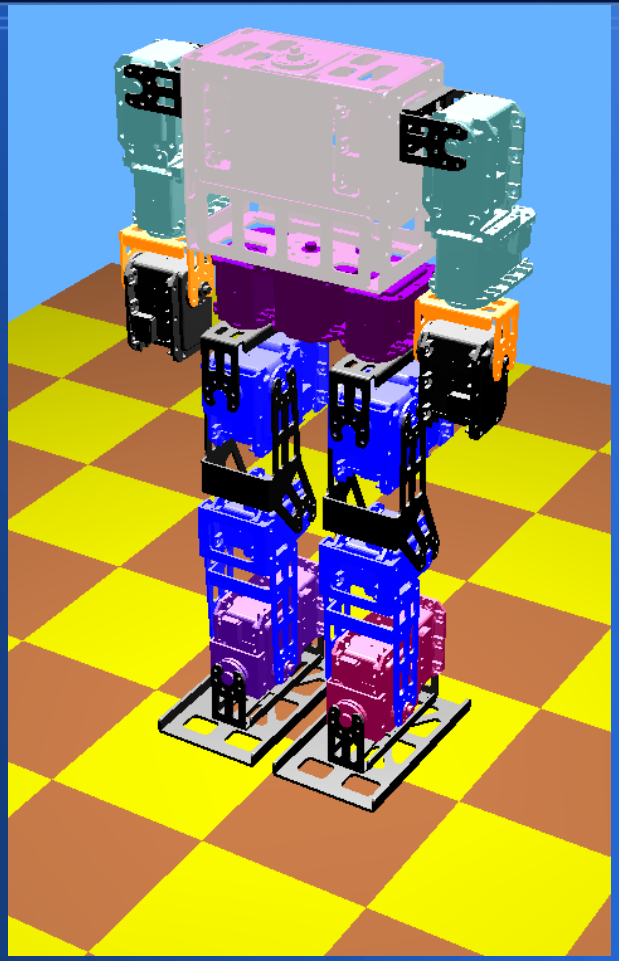


Playback

- Specialize controller can be loaded which reads logged data back from database
- The recorded data can be played back within the simulator
- Simulator provides contact highlighting and single step advance
- Allows for failure analysis

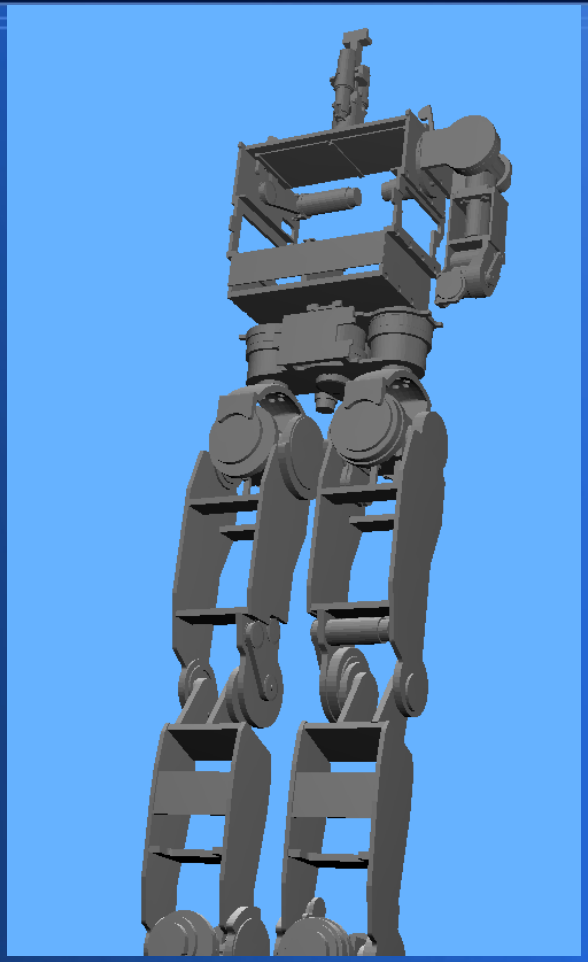
Current Progress

- Miniature Humanoid physical properties fully implemented in simulation
- Prototype software connected to playback simple statically stable walking scripts
- Requires addition of sensors



Current Progress

- HUBO model ~70% complete
- Physically geometry mostly imported
- Appropriate inertial data collected
- Sensors need to be implemented



Movie Time