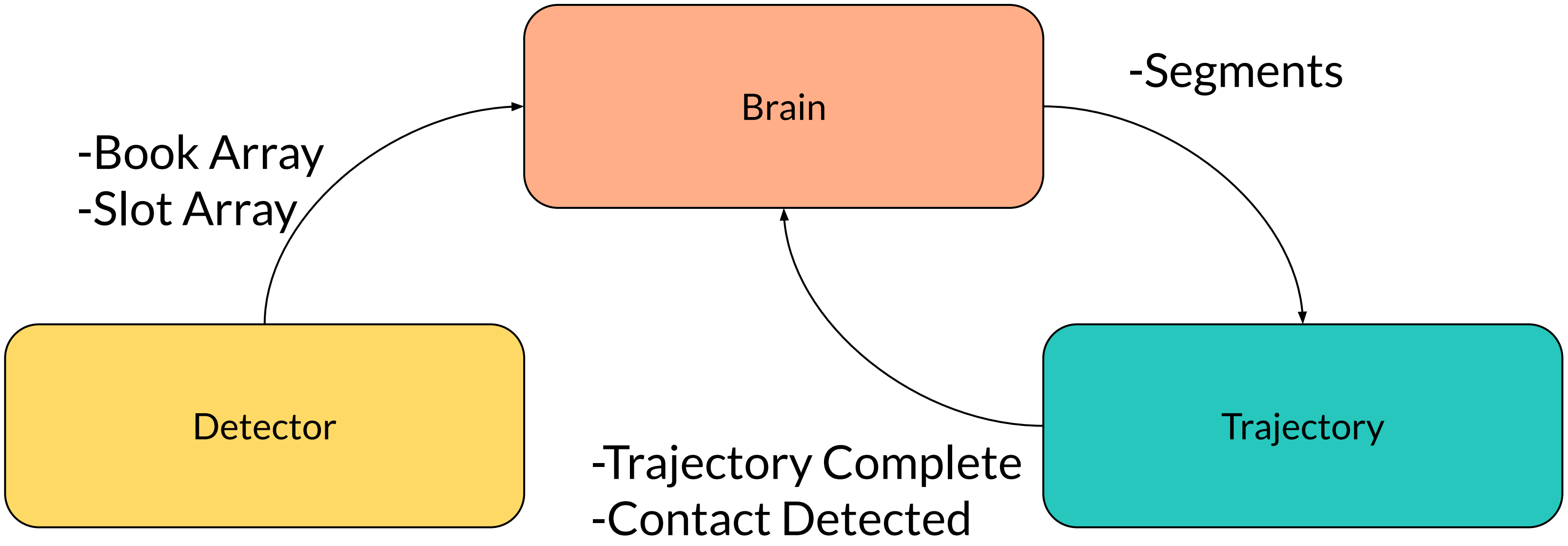
Shelvin' Cooper

ME/EE/CS 134 Project by Eloise, America, and Davis

Software Architecture



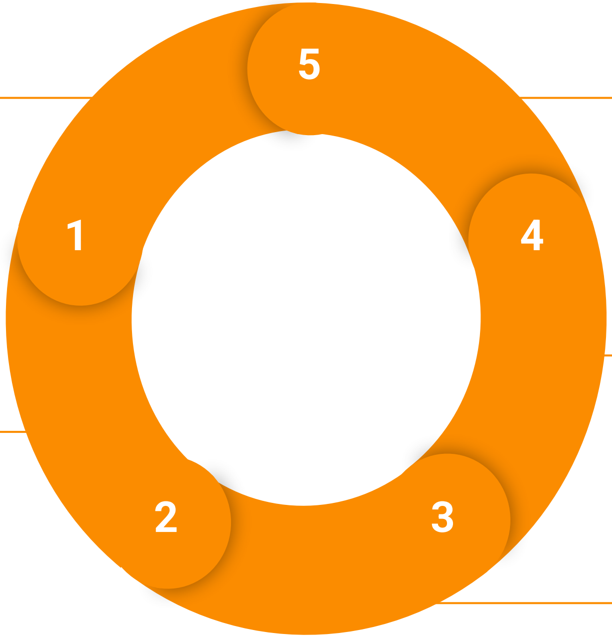
Robot States:

- Init
- Waiting
- Moving
- Contact

State Updates:

- New segments and trajectory complete messages lock after first being published
- Robot moves so that the marker is visible before recalculating the trajectory

Nominal Behavior



Start at Home Position
The robot returns to its home position between shelving books. Robot chooses the book that is closest to one of two pickup zones.

Gripper Chain

Reorient
The robot aligns the spine of the book to be parallel with the edge of the table

Bracket Chain

Translate
The robot pushes the book to the pickup position

Bracket Chain

Pickup
The robot completes the pickup maneuver at the edge of the table

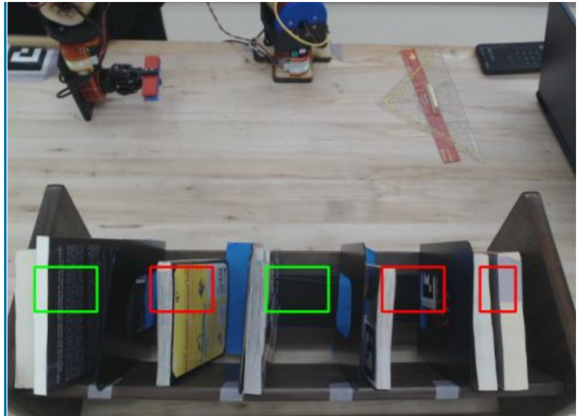
Gripper Chain

Shelve
The robot moves the book over to an empty shelf slot and places it

Gripper Chain

Vision

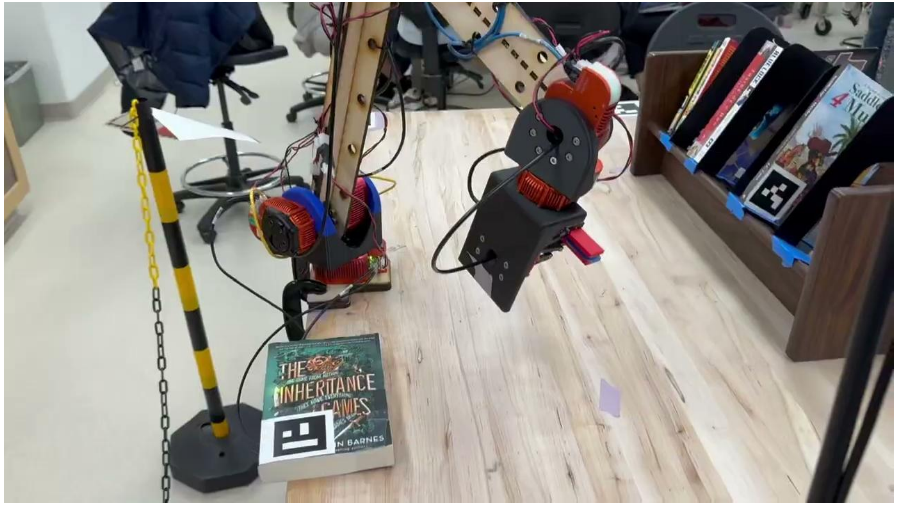
- Intel Camera: Above the workspace
  - Proposed: Detect height of books from the surface of the table.
  - Actual: RGB camera detects ArUco markers on books. Need to locate book's spine.
- USB Camera: Above the workspace
  - Proposed: Determine empty slots with ArUco markers or using OpenCV
  - Actual: Manually input the bounding box of each slot. There are five slots. Compute the mean value.



Recovery Behavior

- Contact Detection
  - Trajectory is replaced with “lift tip” and “return to home”
- Environment Detection
  - Robust to unexpected changes because the book location and orientation is always re-evaluated before making the next step (reorienting, translating, or shelving).
- The robot remains in home position if all book spaces are occupied

Demo



Lessons Learned

- URDF
  - onshape-to-robot: ensure robot is in zero position before converting to URDF
  - Use viewurdf.launch.py to ensure robot behaves the same in sim and hardware
  - If there's a mismatch between transformation frames (specifically orientation), type 3.1415926535 instead of 3.14. Newton Raphson requires this to converge with 10^-6 error, rather than 0.01 error.
- Newton Raphson
  - Initialize robot in home position.
  - If we command robot to go to shelving position, make sure it first hallucinates moving in the x direction, then the y direction.
  - Use the last (hallucinated) joint position as the initial guess for next joint position.
  - Otherwise, the robot collides with the table or itself.

Lessons Learned

- Weight issues
  - Limit workspace of robot
  - Robot can pick up book
  - Use slow movements to avoid resonance
  - Might need stronger structural support so arm doesn't wobble as much
  - Initial design emphasized being light, but we might have had more room to work with than expected
- Error Mapping
  - Robot is usually 0-2 cm off.
  - Error depends on bracket orientation
  - If robot starts from different configurations and is commanded to go to the same point, the robot tip might end up in different places.
  - Need offset.
- Sending lots of messages between nodes requires robust handling
  - Lock out messages (besides warnings) when they are not expected

Thank you!