MONTEREY BAY AQUARIUM RESEARCH INSTITUTE

ESP Puck Tracking

4/12/24 Brent Roman   brent@mbari.org

Why is Puck Tracking important?

- There are many types 2G ESP of pucks
- Pucks must be processed in prescribed sequence
- Dropping a puck ends a deployment
    - typically jambs Elbow or Forearm
- Can flood the can if dropped puck not detected
    - and clamp closed without a puck loaded in it
- We want to detect puck errors early to avoid this!

Why is puck tracking difficult?

- ESP cannot distinguish among puck types.
  - cannot read etched serial #, etc.
  - Even the flush puck appears identical
- No direct sensors for puck presence
  - ESP can’t tell if Hand is holding a puck
  - Opening Hand risks dropping it
- ESP power may be remove at any time
  - even while handling pucks :-(

Sensing Puck Presence

- Motor current when closing Clamp
  - increases as spring compresses
    - only if there's a puck in the clamp!
    - anything puck in clamp sized would do
- IR beam under camera
  - blocked by top puck in stack
    - Elevator position at that instant allows ESP to calculate height of puck stack.
    - Assuming pucks are of standard height.

Counting Pucks

- Expected puck stack height in a tube
  - updated when puck is added or removed
- Every time puck stack lifts, its height
  - is compared with its expected height
  - Puck::Error raised if height != expected
    - within 0.5 puck height units
- This same error will occur if an operator
  - manually changes a tube’s stack height
  - unless the system is informed

Tracking Pucks in play

- Pucks currently in play are tracked from
  - tube stack or garage to Hand to destination Clamp
  - unclamping, back to Hand, to tube stack or garage
- Each puck move is recorded in $ESPmode.slot file
- When ESPserver starts,
  - it replays the .slot file to determine location of all pucks
  - If any are out of place (i.e. in play)
    - – ESP.ready! will avoid dropping pucks
      - reclamps previously clamped pucks
      - Hand refuses to open if it previously held a puck.

Resetting Puck Tracking

```
-> Puck.count      #counts all pucks in carousel
-> Puck.count 2    #count only pucks in tube 2
-> access 2        #present tube 2 and forget its height
-> access          #present currently selected tube, forget its height
```

- These do not move any actuator:

```
-> clear!          #forget all puck heights
-> clear! 2..4     #forget puck height of tubes 2, 3 and 4
```

- If you see any of these sorts of messages on startup:

`Storage::Warning in simfast – (`*`clamp`*`|Hand).holds *`*`puck`*

  - the system thinks that pucks were in play when it last exited
  - If you have already manually corrected this situation,

```
-> Storage.recovery.clear    #forgo automatic puck recovery
```

  - ***BEFORE*** executing -> `ESP.ready!`

Puck Loading for Simulations

-> `pucks` #returns # pucks in each tube as a Hash

-> `pucks 2=>14, 3=>9` #set count of pucks for just tubes 2 & 3

-> `fill!` #fill carousel tubes 2..7 with pucks, empty others

- `TubeCapacity` = 22 (or 33 if using “short” pucks)

-> `fill! 14, 2, 3` #fill tubes 2 & 3 with 14 pucks, empty others

- To ensure each simulation run starts from tube 2 with a full carousel:

```
if ESP.simulation?
   fill!
   startTube 2
end
```

Put the above before the mission