**Generate TRUEMAP**:

1. teleoperate turn 30 deg (variable) for 360

* Don't start SLAM yet (just don’t press enter yet) -> self.ekf\_on = False
* press ‘**m**’ to tell it we have reached first 30 deg (hardcode pose), save measurements (+ taglist) of all markers u can see into array (taglist and rel x,y dist)
  + Make sure when teleoperate robot is turning left (+ angle)
* Once completed 360,
  + Take 4 random markers and tags from each quadrant
  + Save into TRUEMAP.txt
  + **BONUS**: take 1 measured marker from each quadrant (extra 1 bonus)
  + Reset robot theta to 0

2. do M2 SLAM

* Run ekf.py (original from M2)
* Press ‘**s**’ to save slam map -> SLAM.txt AND align markers (step 3) AND go to next mode (step 4)

3. use 4 points from 1 to align 10 aruco estimations

* SLAM.txt + TRUEMAP.txt -> TRUEMAP\_ALIGNED.txt

4. Don’t close operate.py, reset robot pos to (0,0,0), switch to ekf2.py (covariance = 0) (all parameters will reset), put truemap markers into self.markers and truemap taglist into self.taglist

5. do SLAM teleoperate again, find fruit positions using (hopefully correct robot pose) & target pose est -> targets.txt

* Find fruit
* Turn 360 if needed to improve robot pose
* press '**p**' to detect image once
* press '**n**' to save lab\_output/pred\_x.png, robot pose + raw img filename in lab\_output/images.txt and raw image in pibot\_dataset/img\_x.png
* Look around until pose is good (prepare start point for M4)
* Press ‘**b**’ to run TargetPoseEst : images.txt -> targets.txt
* Update TRUEMAP.txt with found fruit positions

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Autonomous fruit search**:

* Pause SLAM, Continue from last position from M3
* only after drive
* For each fruit:
  + Generate path
  + For each waypoint in path:
    - Turn to next waypoint angle
    - Drive straight to waypoint
    - Update ‘desired pose’
    - Resume SLAM and do 360
    - update robot pose from self.ekf.robot into fruit\_search.robot\_state
    - If robot pose from SLAM far off from desired pose (set threshold), attempt to drive to waypoint again
    - Do SLAM 360 to check again
    - update robot pose from self.ekf.robot into fruit\_search.robot\_state
    - Use an aruco to ground robot pose
    - Pause SLAM

**Things to do**:

* **~~Astar:~~**
  + reduce resolution
  + make it possible to take any random location on map and map into grid box in astar
  + Cut out some points for each waypoint
  + Make adjacent boxes around obstacles as obstacles also
* ~~Recalibrate camera~~
* ~~Recalibrate wheels~~
* ~~Check robot pose accuracy when searching for fruit~~
* ~~test measurement of single marker (is it accurate/ need to add constant to calibrate it)~~
* ~~Combine both ekf into 1 file~~
* ~~Fix singular matrix error~~
* ~~Modes:~~ 
  + 1: initial 360 measure markers
  + 2: M2 SLAM
  + 3: M3 search fruit pos
  + 4: M4 autonomous
* ~~Fill up remaining measured markers if got any missing from any of the 4 quadrants~~
* ~~Chg mode 1 to autonomous~~
* ~~In mode 1, pressing front and back will move the robot back and forth very tiny bit, also code turning right in case overshoot~~
* ~~Divide big turning angles into sections of 30 deg~~
* ~~Recalibrate fruit scale in target pose~~
  + Plot out relation
* ~~Fix and mark center of robot (screw on camera)~~
* ~~Calibrate 30 degrees for 360 (lv, rv)~~
  + Do separate lv rv scale for each of the 4 movements
* ~~Code & Test mode 4 autonomous~~
* ~~Omit waypoint correction step~~
* ~~Test on different arenas (and choose preferred one)~~
* ~~Make robot 360 overshoot rather than undershoot~~
* ~~Reduce delay and see if everything still works~~
* ~~Still need to improve marker measurement calibration?~~
* ~~Find a way to estimate 30 degrees for initial 360 accurately~~
* ~~Automatic pause and start slam for mode 2 & 3~~
* ~~Leave delay only in drive and turn~~
* ~~Test mode 1 with map where one of the quadrant has no markers, test if code is robust~~
* ~~Test stopping halfway at every mode and resume from latest mode~~
* ~~Do plot for CV\_eval and use it after press "b"~~
* ~~Plot initial measured markers~~
* ~~Test M3 no SLAM~~
* ~~Try M3 with turn 360 measure, not SLAM~~
  + Drive to a point where robot can see both markers and a fruit at same time
  + Press “m” to turn360 measure but only get robot x y + Change mode to small steps mode
  + When press “left” or “right”, will turn in small\_angle
  + When reach desired theta (make sure can see fruit & at least 1 marker at same time), press “m” to measure theta, “p” to get detection, “n” to save prediction, pose and raw image, “b” when complete all fruits
* ~~Test with markers and fruits not at intersection~~
* ~~Time management~~
  + Mode 1: 1 min
  + Mode 2: 4 mins
  + Mode 3: 5 mins
  + Mode 4: 10-12mins
* ~~Tune self.small\_dist~~

**Extra:**

* Do GUI for autonomous?
* Do clayton wheel test and obtain R\_factor, Q\_factor (covariance)
* Use computer vision at a certain waypoint to go straight to fruit and visit fruit
* Use arucos to align robot when driving straight?
* ~~Use aruco to align for turning?~~