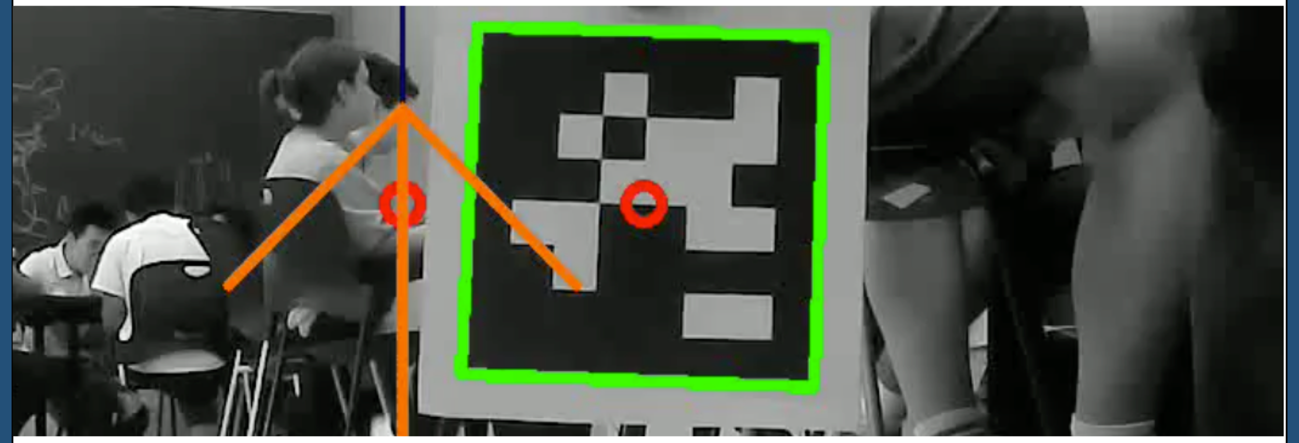
# Autonomous Underwater Vehicle Challenge

Joshua Zyzak, Travis Tran, Arjun Chavan, Kevin Yao, William Gao

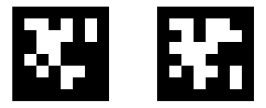
Advisors: Dr. Mohamed Saad Ibn Seddik, Max Berman, Vincent Vandyck, Syed Mohammad Asjad

# Autonomous Decision Making

## AprilTags Detection



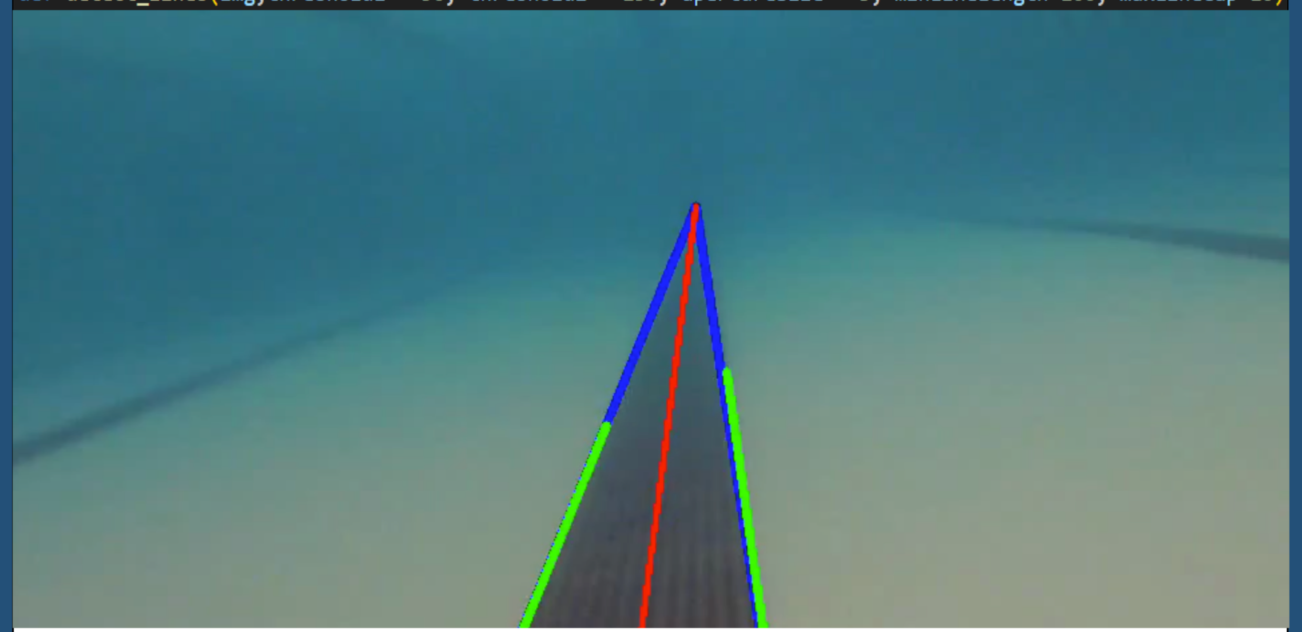
The AUV uses AprilTags to determine its position relative to other AUV's. This information is then fed into the PID and used to chase (predator) or run away from (prey) the other AUV.



```
at_detector = Detector(families='tag36h11',
                       nthreads=1,
                       quad_decimate=1.0,
                       quad_sigma=0.0,
                       refine_edges=1,
                       decode_sharpening=0.25,
                       debug=0)
```

## Pool Lane Detection

```
def detect_lines(img,threshold1 = 50, threshold2 = 150, apertureSize = 3, minLineLength=100, maxLineGap=10)
```



The AUV uses computer vision edge detection to estimate the position of the lane. Using this, the AUV can orient itself in the swimming pool to align with the lane's center. Green lines represent where the AUV found lines, and the blue lines represent where the AUV thinks the lanes are. Red represents the center of the lane.

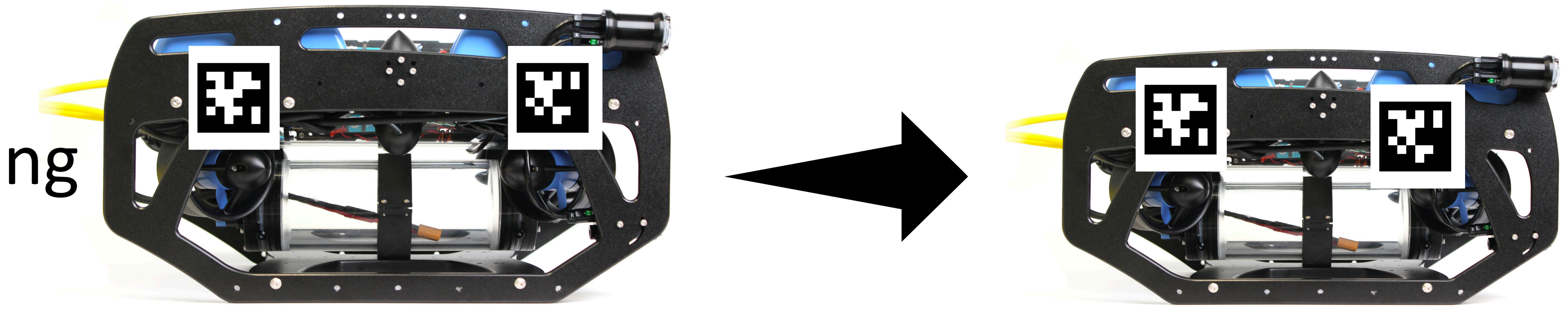
## End Goal: Autonomous Hunter and Prey

After implementing an algorithm for autonomous decision-making and motor control in the AUV, we developed a game to test these systems.

The "hunter" robot will navigate the pool by following the pool lanes until it sees the prey robot. Once it recognizes the prey robot using April Tags, it will then chase it and flash its lights if the prey is "hit"

The "prey" robot will navigate the pool using the lane lines until it sees the hunter robot. When it does, it will use AprilTags detection to know where to run away.

Chasing



Running Away

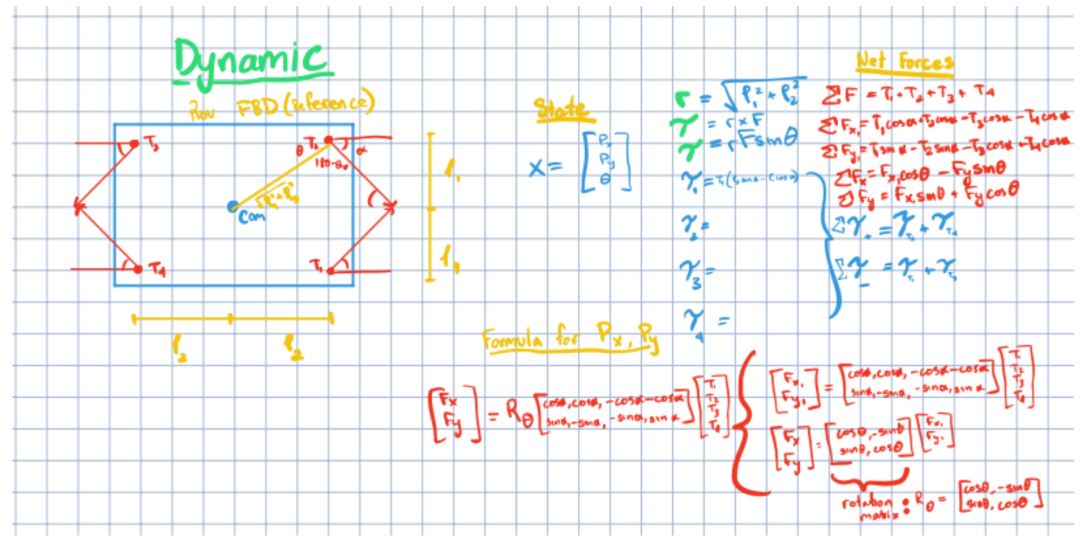
## Challenges Along the Way

- Computer vision and autonomous decision making proved to be technically challenging.
- The very limited pool time we had meant that we were often running lots of untested code.
- Communicating with the AUV and integrating our code with the existing AUV protocols.
- Coordinating times to meet and share code with each other.

# Autonomous Control

## Kinematics

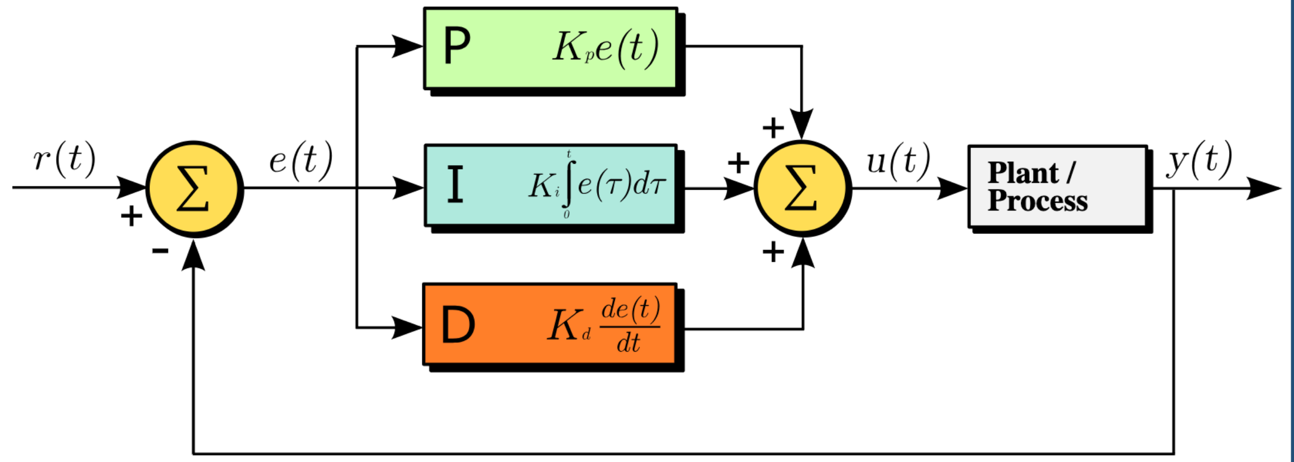
$F_b = \rho \cdot V \cdot g$



Kinematics allows the AUV to know how much power should be given to each thruster given a desired heading and direction. This allows the AUV to move independently in 4 degrees of freedom.

## PID control

$I = \sum_i m_i \cdot r_i^2$




After choosing a desired point, the AUV will navigate to it using PID. The PID controllers then calculate the error from the desired point and the first and second derivatives of this error to determine where the AUV should.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$