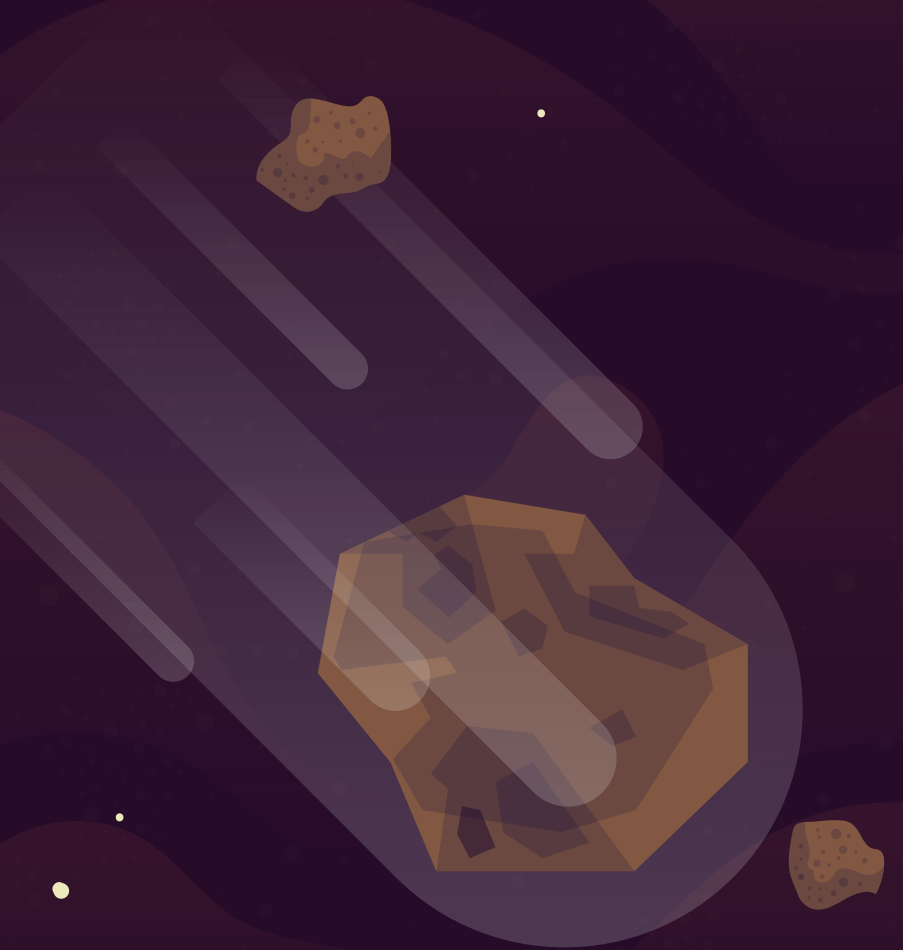


Space Object Detection and Interception

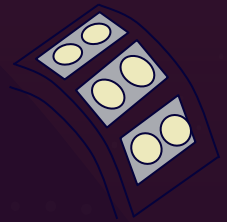
Final Demo

Muaz Shash, Youssef Elhadad, Damian Pacynko, Marko Ciric
Team 36

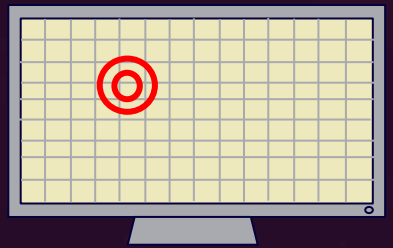
Danger from Above!

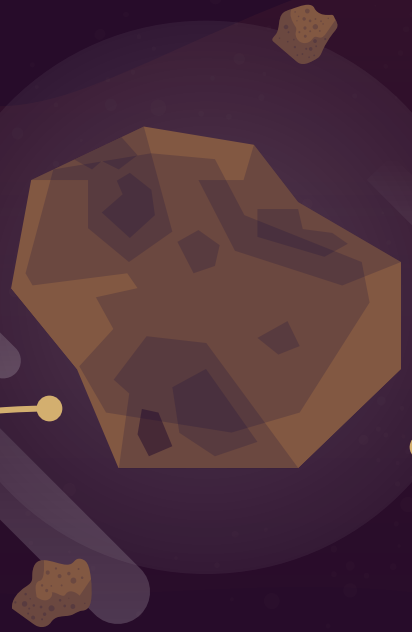
Our (Space) Mission

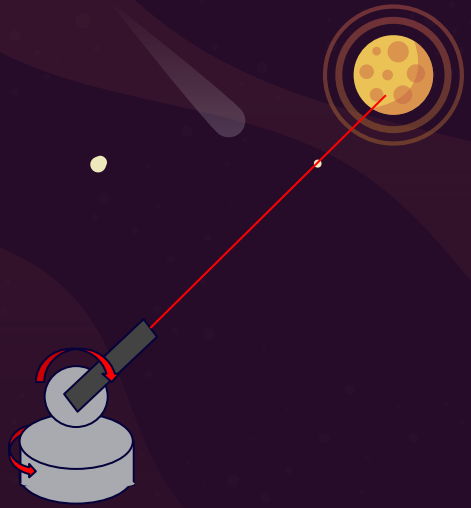
1. Detect



2. Predict

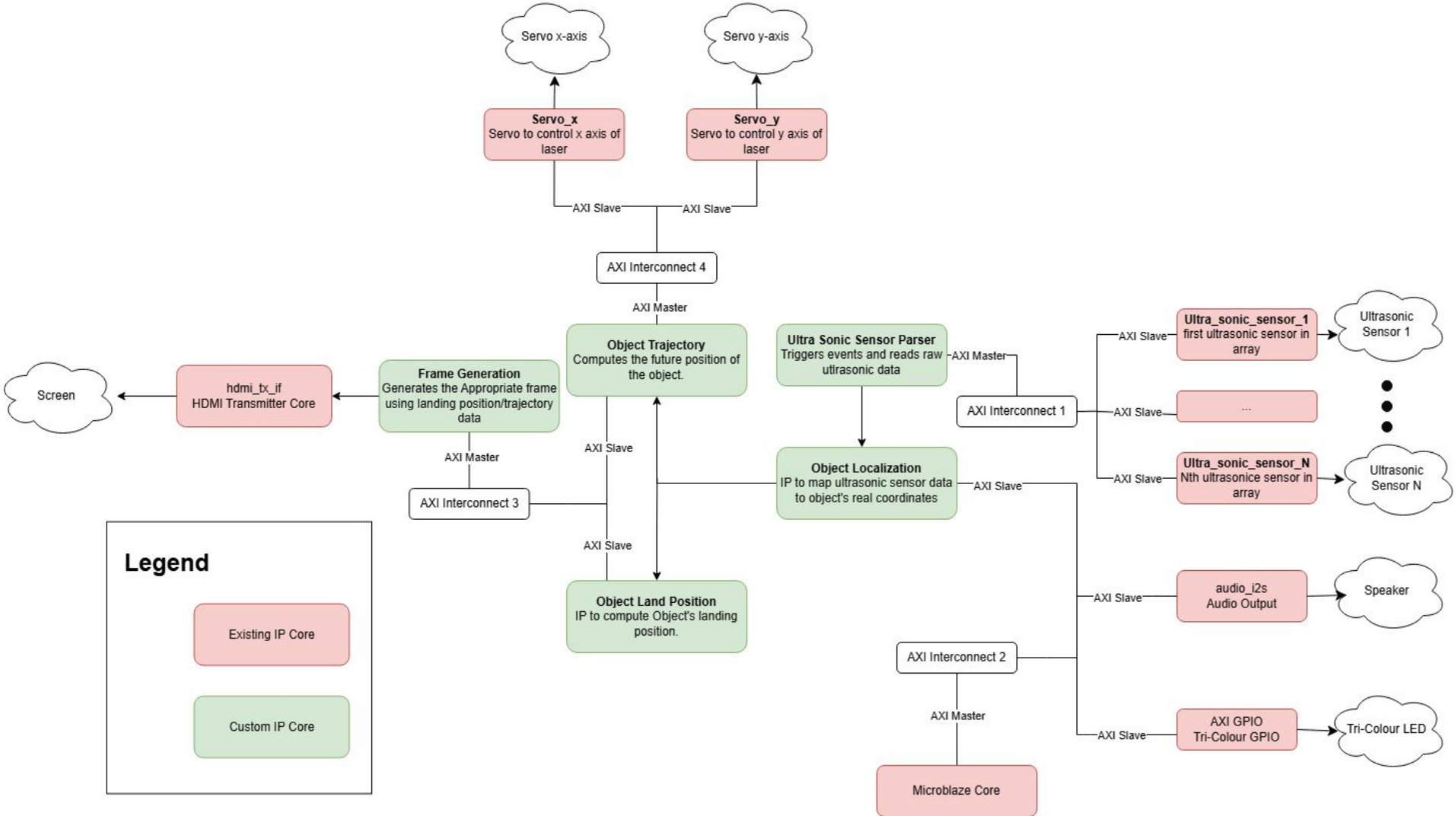




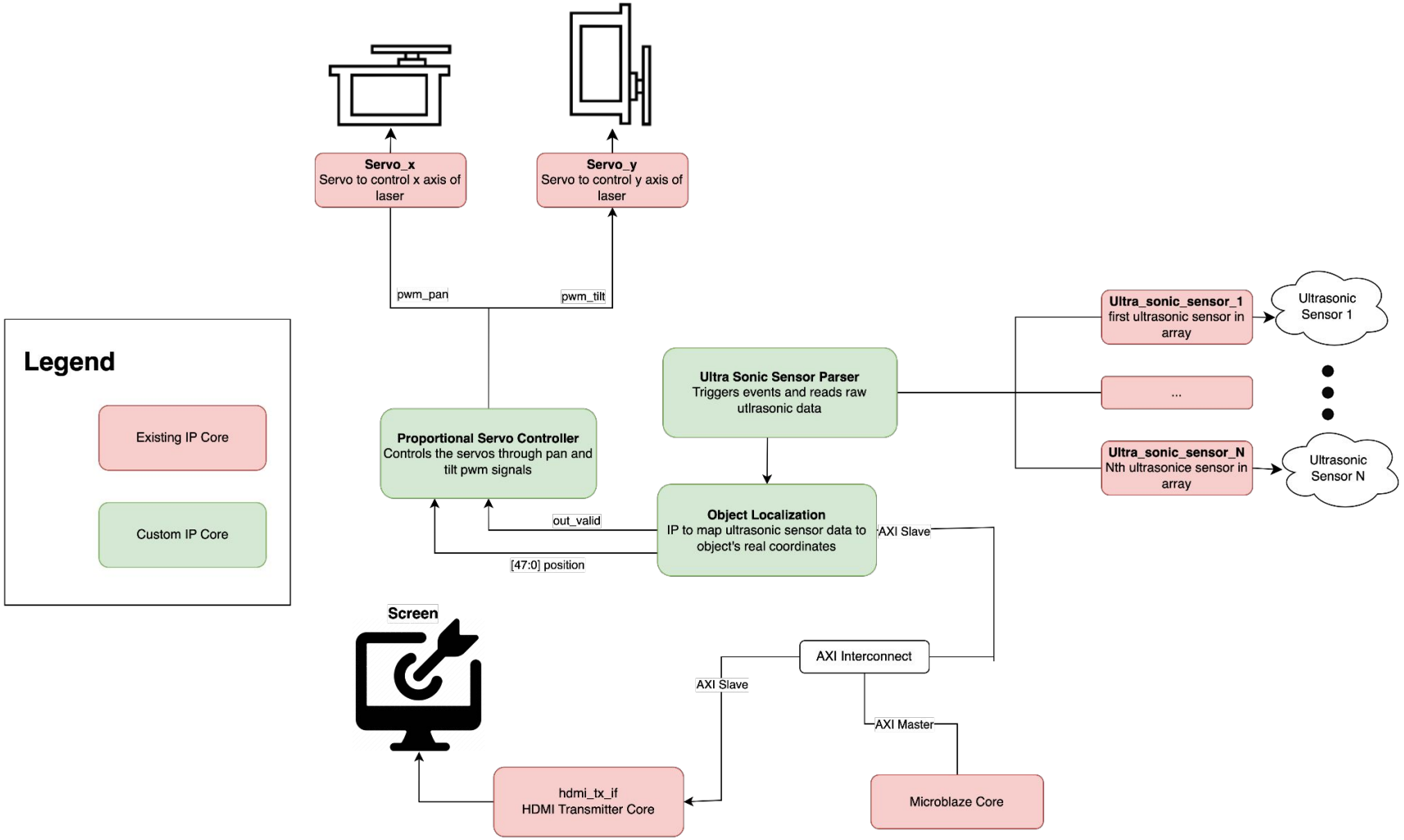


3. Intercept

Initial Proposed System


Servo x-axis
Servo y-axis
Servo_x
Servo to control x axis of laser
Servo_y
Servo to control y axis of laser
AXI Slave
AXI Slave
AXI Interconnect 4
AXI Master
Object Trajectory
Computes the future position of the object.
Ultra Sonic Sensor Parser
Triggers events and reads raw ultrasonic data
AXI Master
Ultra_sonic_sensor_1
first ultrasonic sensor in array
Ultrasonic Sensor 1
AXI Slave
AXI Interconnect 1
AXI Slave
...
AXI Slave
Ultra_sonic_sensor_N
Nth ultrasonice sensor in array
Ultrasonic Sensor N
Screen
hdmi_tx_if
HDMI Transmitter Core
Frame Generation
Generates the Appropriate frame using landing position/trajectory data
AXI Master
AXI Interconnect 3
AXI Slave
AXI Slave
Object Localization
IP to map ultrasonic sensor data to object's real coordinates
AXI Slave
Object Land Position
IP to compute Object's landing position.
AXI Slave
audio_i2s
Audio Output
Speaker
AXI Interconnect 2
AXI Master
Microblaze Core
AXI Slave
AXI GPIO
Tri-Colour GPIO
Tri-Colour LED
Legend
Existing IP Core
Custom IP Core

Final System


Legend
Existing IP Core
Custom IP Core
Servo_x
Servo to control x axis of laser
Servo_y
Servo to control y axis of laser
pwm_pan
pwm_tilt
Proportional Servo Controller
Controls the servos through pan and tilt pwm signals
Ultra Sonic Sensor Parser
Triggers events and reads raw utlrasonic data
Ultra_sonic_sensor_1
first ultrasonic sensor in array
...
Ultra_sonic_sensor_N
Nth ultrasonice sensor in array
Ultrasonic Sensor 1
Ultrasonic Sensor N
Object Localization
IP to map ultrasonic sensor data to object's real coordinates
out_valid
[47:0] position
AXI Slave
Screen
AXI Interconnect
AXI Slave
AXI Master
hdmi_tx_if
HDMI Transmitter Core
Microblaze Core

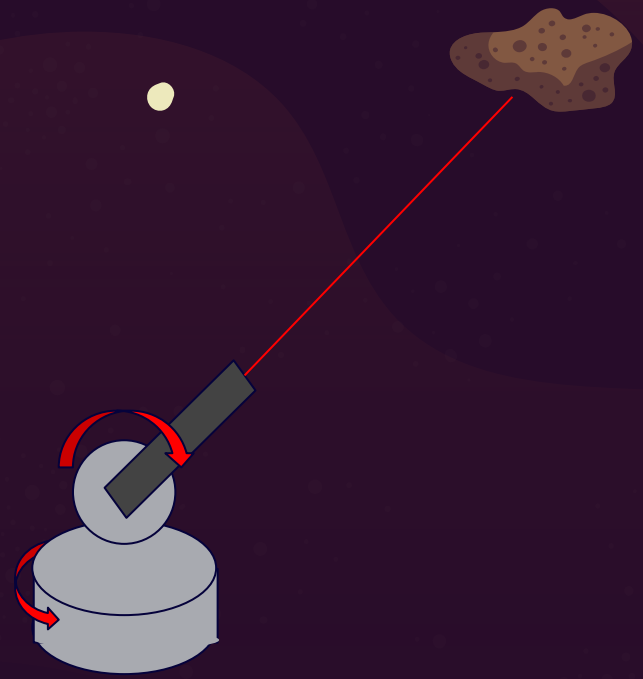
Diving Into the Details

Our Custom IPs

1. Servo Control
2. Sensor Data Parser
3. Localizer

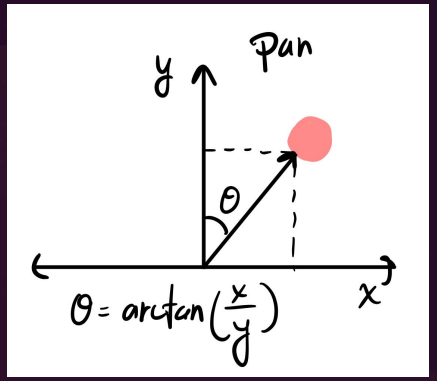
Servo Control

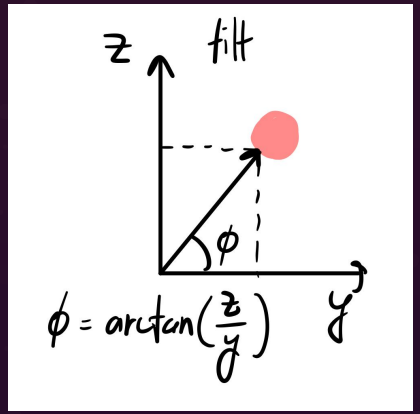
- Moves laser to point at position specified by localizer
  - Servos use “pan and tilt” configuration, controlled by pwm
- Controller that increases pulse width proportional to desired angle



Servo Control - Calculation

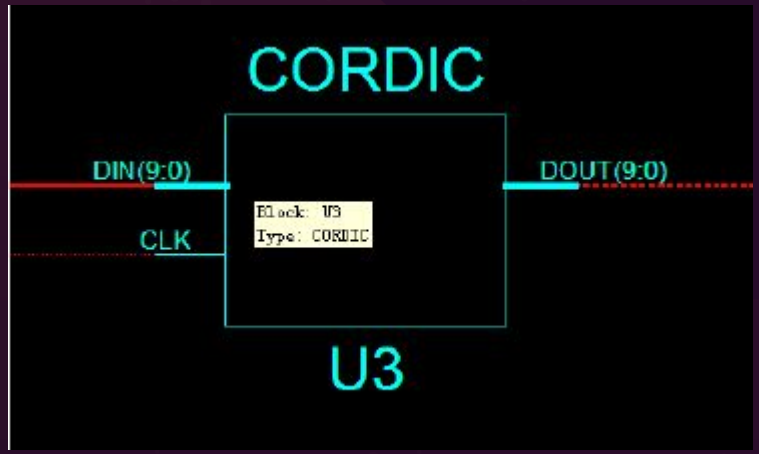
- Need to find the pan and tilt angles.
  - **arctan(x/y)** for pan
  - **arctan(z/y)** for tilt
- Translate this angle to a pwm pulse width





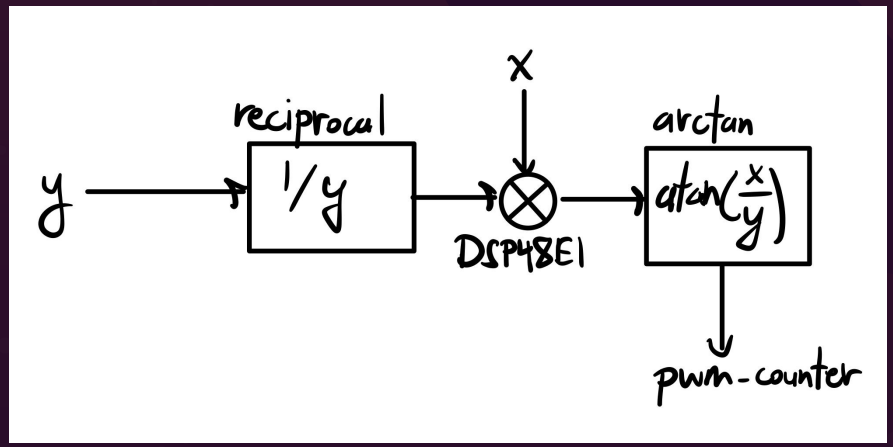
Servo Control - Initial Design

- Initially wanted to use Xillinx CORDIC IP
- Problems:
  - Large latency
  - Complex
  - Difficult to debug
- Solution:
  - Custom implementation
  - Look up tables



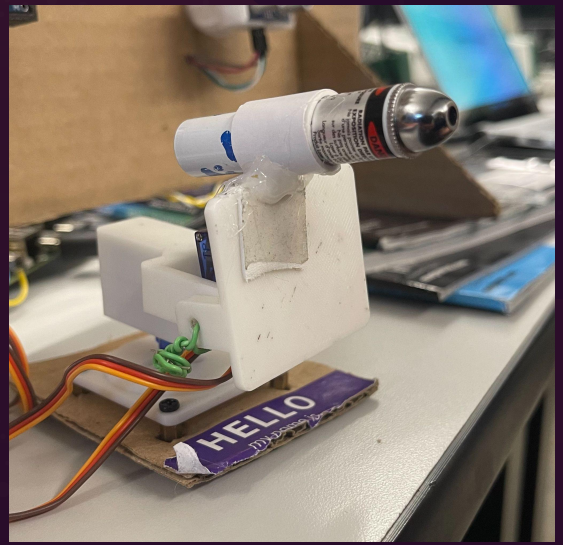
Servo Control - Final Design

- Replace division in arctan argument with reciprocal LUT and multiplication
  - Input is x, output is 1/x in 0.16 fixed-point format
  - Multiply int16 with 0.16 fixed-point
- arctan LUT that translates (x/y) argument directly to pulse width counter



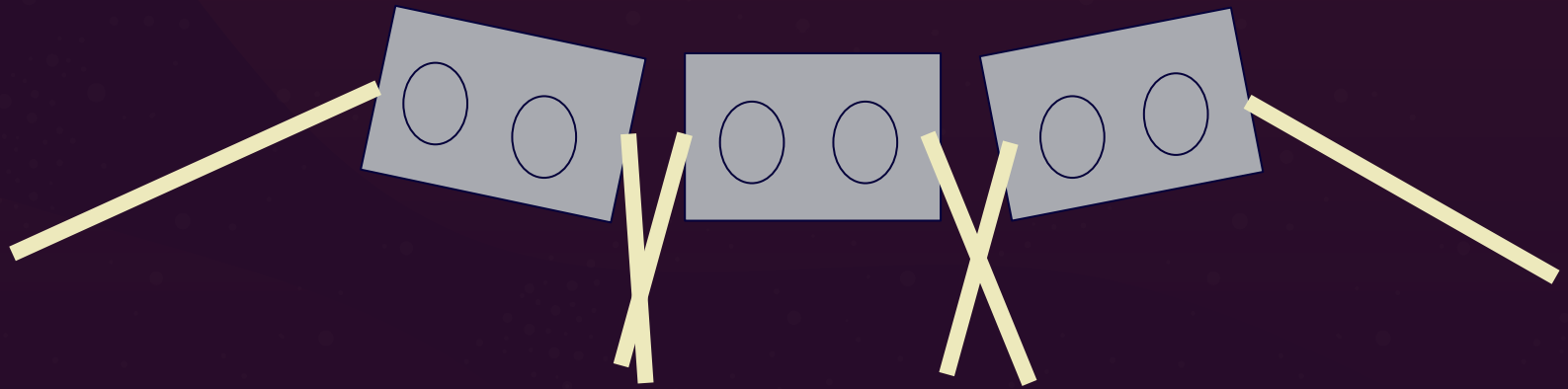
Servo Control - Challenges

- Pan/tilt mechanical design
- Manually calculated angles not matching servo rotation



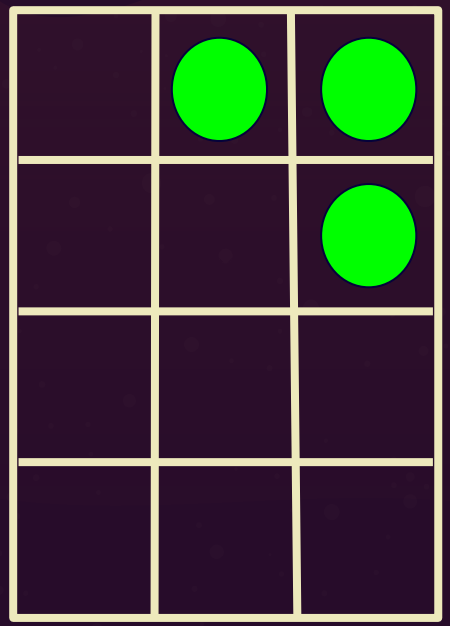
Localizer

- Computes relative X, Y and Z coordinates given an array of distances from sensor parser.
- X = sensor x + distance*sin(sensor_angle)
  Y = distance*cos(sensor_angle)
  Z = sensor z

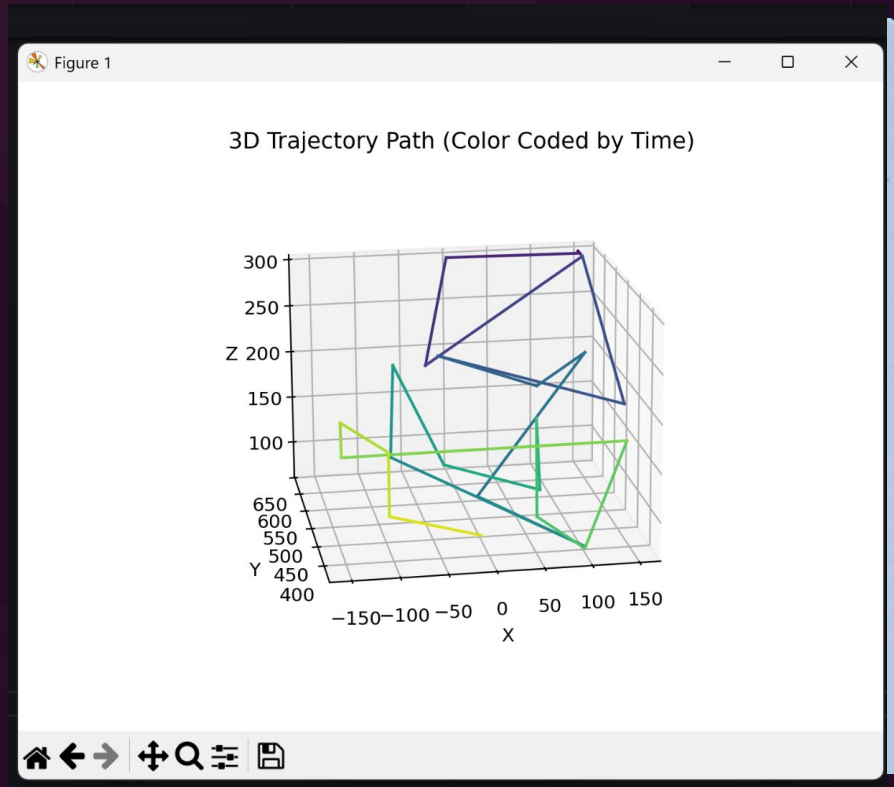


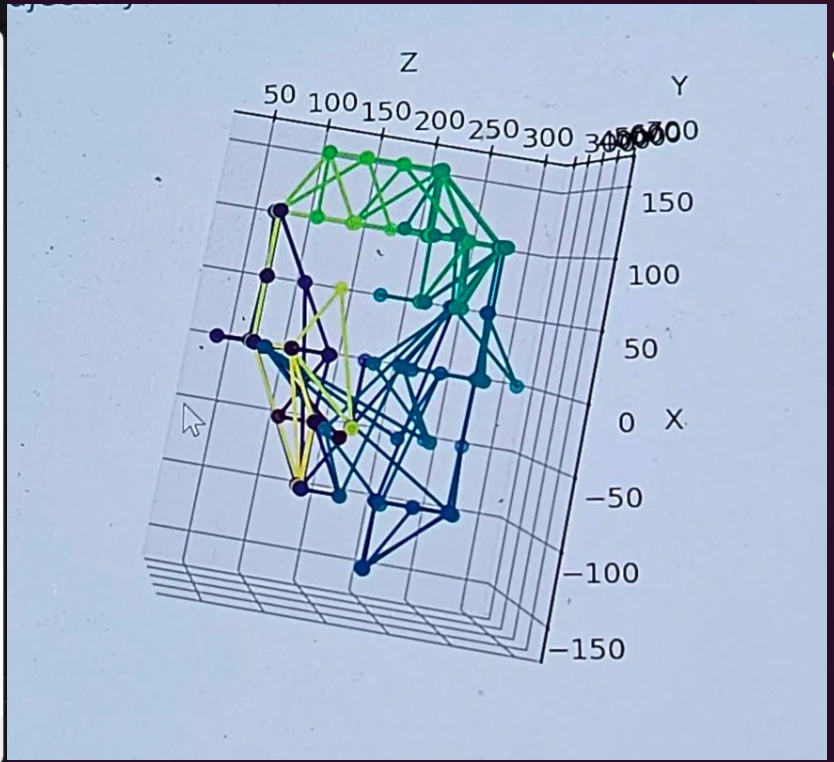
Mk1: Sensor fusion

- Average the output given by the 3 minimum sensors.



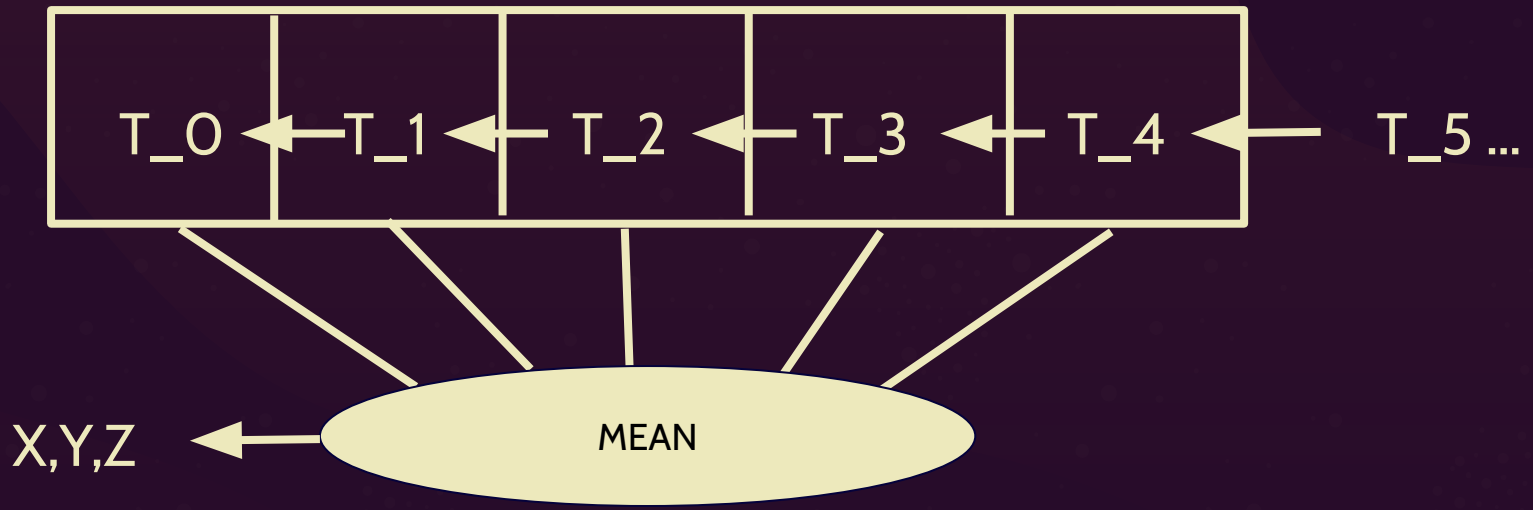
Mk1: Sensor fusion



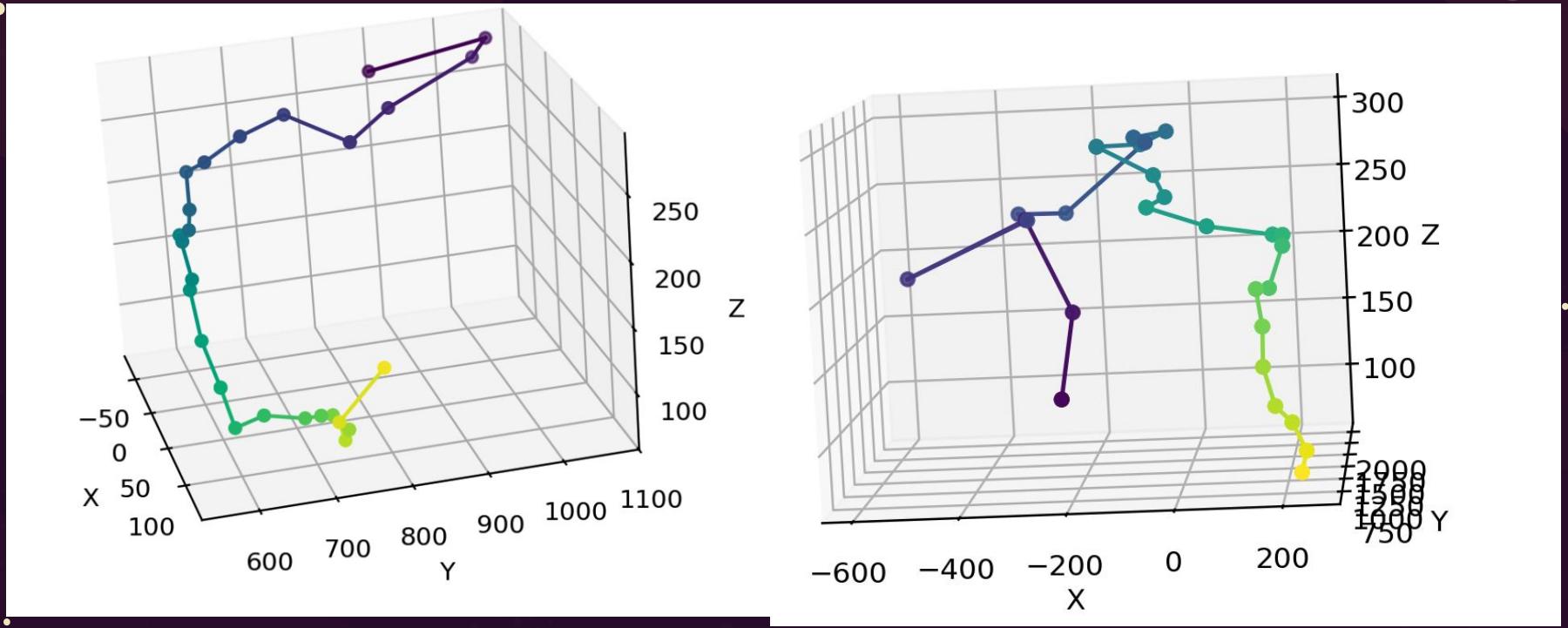


Mk2: Time Averaging

- Average the output of the localizer over multiple time units

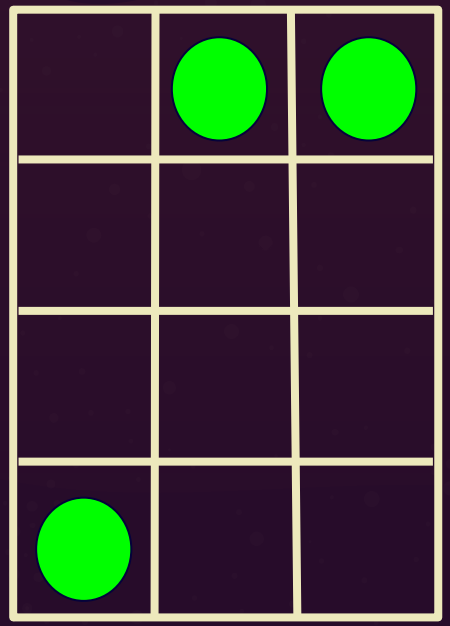


Mk2: Time averaging



Mk3: Nearest Neighbours

- Define sensor fusion based on majority vote and proximity to other sensors.



HDMI Xilinx IP

Button interrupt triggers screen switch between displaying circles on object's location in:

Screen 1: x-z plane and y-z plane
Screen 2: x-y plane

Visualizes results from localization

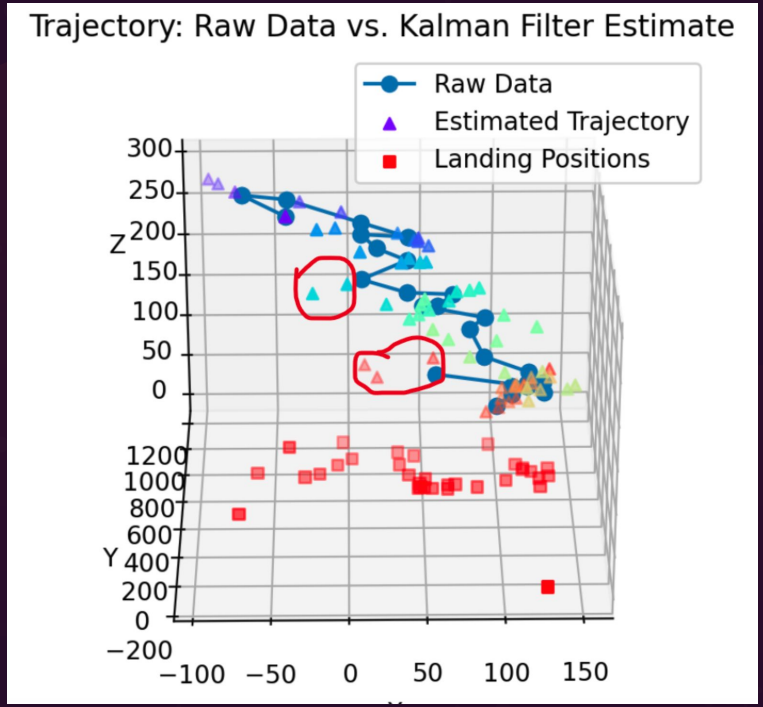
Computed Complexity

Ultrasonic Rangefinder PMOD	0.5
Custom IP	1.5
HDMI Output	1
Visualize meaningful results with a GUI	0.75
GPIO Servo Control	0.2
Total	3.95

Challenges

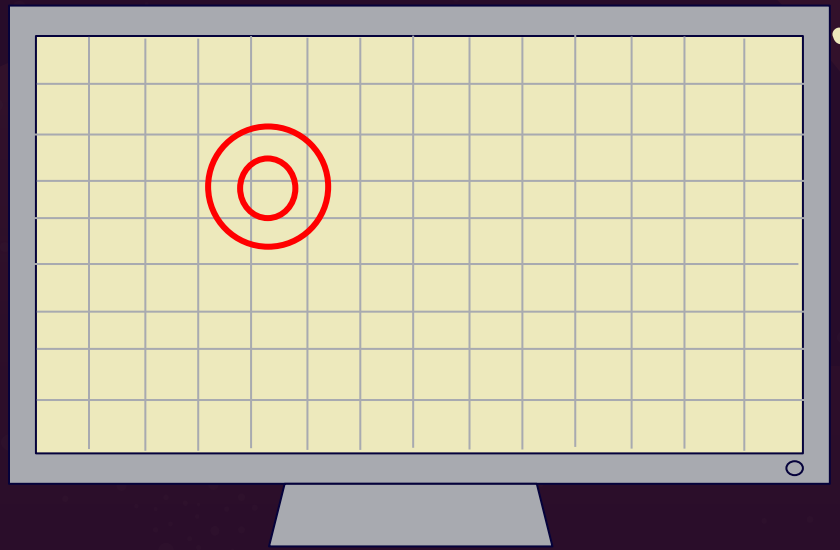
Implementing Trajectory

- Kalman Filter to predict object path
- Inaccurate predictions from IP
- Sensor data too noisy to predict even in a python program
- Shifted to more robust localization

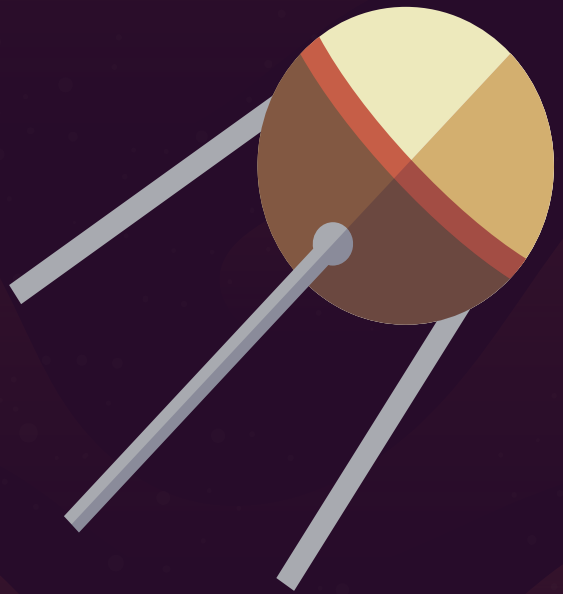


Future Goals

- Develop accurate Trajectory algorithm
- Build robust mechanical package
- Spend more time tuning setup in different orientations and positions for optimal FOV
- Integrate Landing Position tracking



DEMO!



Q & A

Thanks for Listening