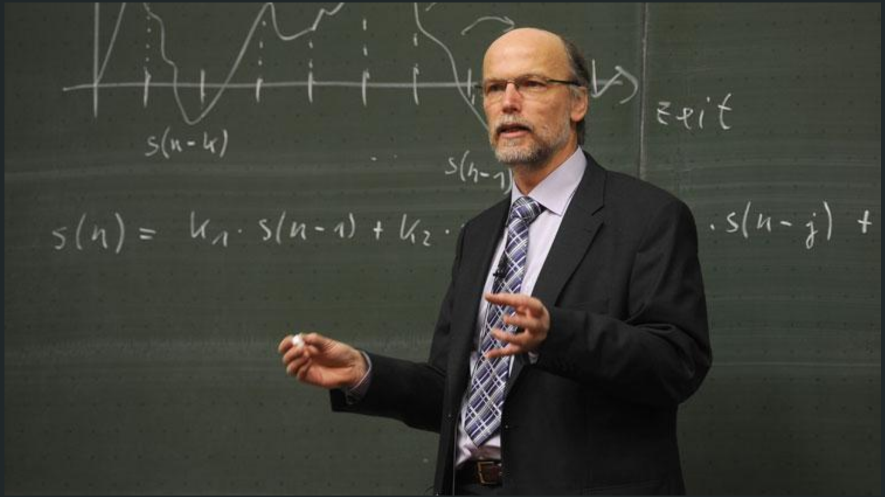
InFrame

Team A5: Diego Martinez, Ike Kilinc, Ismael Mercier

InFrame

Application Area

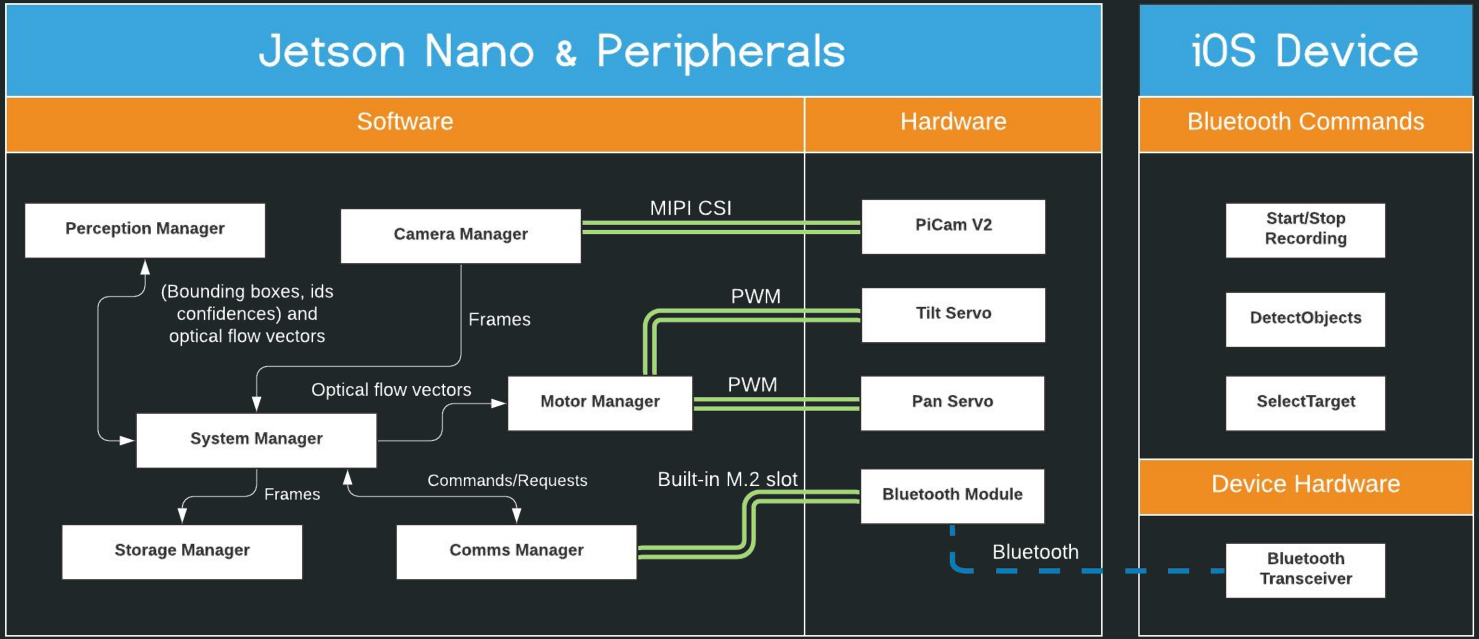
Lecture Filming



Action Shots

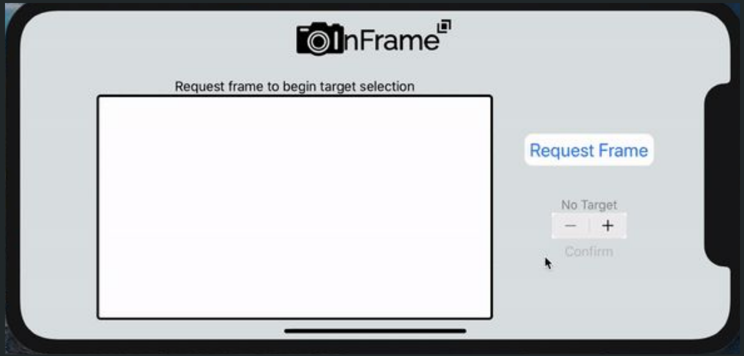
InFrame

Solution Approach



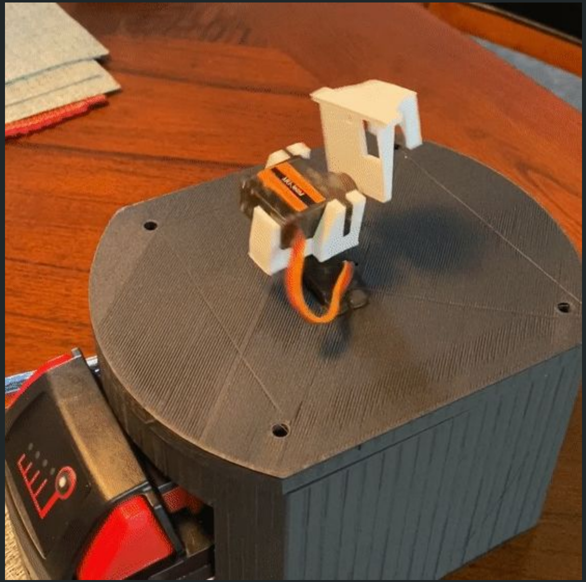
Complete Solution

1. Target Selection



2. Target Tracking

3. Motors Follow Target



Application | **Solution** | M&V | Trade-Offs | Proj. Mgmt.

InFrame

Metrics and Validation → Hardware

Goals	Requirements	Testing Method	Result
Motion Speed	Servos should move at least 40°/s	Record full rotation time w SW timer	Motors are capable of 100°/s
Position Accuracy	Within 10 degrees of target	Moved CW and CCW by the same amount 10 times	Average of 5-7 degree drift
Smooth Motion	Ignore movements < 0.5m	At a fixed user-defined distance, manually verify that small movements do not trigger motors	Software ignores optical flow vectors which magnitude moves the motors to an angle translating to movements of < 0.5m.
Battery powered	Battery life would last the length of a trip (e.g. hike)	Prove theoretically since PCB manufacture not possible	N/A (was not able to test the battery from home)
Backpack storable	Size < 26.5"h x 17.5"w x 6.5"d	Measure with ruler, compare to pre-design quantities, place in backpack	Measurements: 6.25" x 4.5" x 5.5"

InFrame

Design Trade-offs ➔ Hardware

System PCB

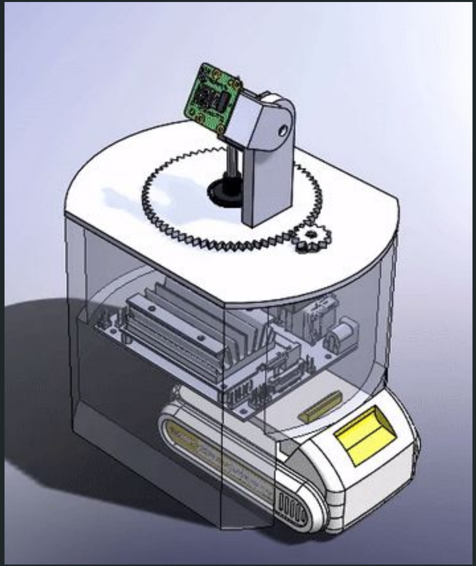
- ➔ Decided to use one PCB instead of two for battery management and motor drivers.
  - ◆ Easier to design and manufacture
  - ◆ Easier to integrate into housing
  - ◆ Less wiring

Voltage Regulator

- ➔ Changed to switching voltage regulator instead of linear regulator
  - ◆ Linear voltage regulator was burning ~46W as heat
  - ◆ Switching regulator is much more efficient

Housing

- ➔ Smaller form factor + increased portability.
- ➔ Cut out continuous rotation due to design complications



Metrics and Validation → Perception

Goals	Requirements	Testing Method	Result
Accurate Object Detection	> 80% accuracy in frames with our use cases	Run on suite of 120 potential targets: 40 lecturers, 40 runners and 40 skateboarders, binary success metric	92.5% accuracy on lecturers 77.5% accuracy on runners 85% accuracy on skateboarders
Tracking Reliability	With no occlusions and no lighting changes, never lose target	frames_tracked / total_frames from suite of 10 tests > 99%, bounding box automatically drawn and manually verified	6 "easy" tests = 718/720 4 reach tests = 417/480** **Includes occlusions, lighting changes and shape changes!
Track Any Possible Target	Identify any target and be able to differentiate between any pair of targets	Select different range of targets and compare with benchmark above	Can only track one of any detection class in a frame

Design Trade-offs ➔ Perception

Object Tracking Implementation

- ➔ **MOSSE Tracker:** As fast as it gets, not very good.
- ➔ **CSRT Tracker:** *Very* slow, supposedly the best but still not great.
- ➔ **MOSSE + Intermittent Obj Detection:** Almost as fast as standalone MOSSE, works very well.
- ➔ **Obj Detection + TensorRT:** Incredibly fast, works very well. A bit more jitter in optical flow but manageable.

Tracking Type	Speed	Targets Lost
MOSSE Tracker	5.98 FPS	19/20
CSRT Tracker	1.19 FPS	14/20
MOSSE Tracker + OD	4.83 FPS	1/20
OD + Inference Boosts	13.75 FPS	0/20

Object Detection Model / Compute

- ➔ **MobileNet-V2 on board:** Fast but light (39 FPS)
- ➔ **ResNet-18 on board:** Very slow on Jetson (5 FPS)
- ➔ **Heavy Models on AWS:** Much faster than necessary, limited by ping to closest AWS servers (Approx 20 ms)

Metrics and Validation → CSM

Goals	Requirements	Testing Method	Result
Minimizing latency overhead	Inter-frame operation time falls within the 60ms threshold of negligible mechanical reaction time	Run the CSM independently of Perception, with frame data mock-inserted into execution flow	13.75 FPS perception (72.7ms), 4.13 FPS full CSM (242ms), CSM adds 170ms per frame on avg
Responsive target selection	User-input response time ≤ 2s	Manually time each possible Bluetooth message transmission to the CSM: timeSentFromRI - timeReceivedByCSM	**243 ms** 43ms CSM processing time + 200ms BT latency (average)
Robustness	System execution cannot be ended abruptly by any possible user-input	Give app to 3 non-developer users, ask them to try and break it	3 major errors found in first test, All 3 were fixed, 2^{nd} test **could not be broken**.

Design Trade-offs ➔ CSM

Control Interface

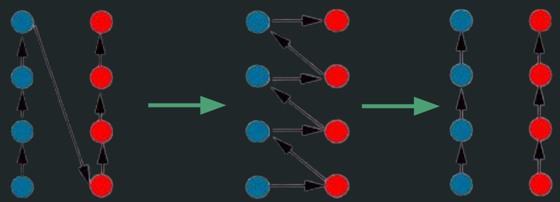
- ➔ On-Board ➔ Remote Interface: Reducing cost & improving convenience
- ➔ Web-Server ➔ Android ➔ iOS: Cutting down on user-input latency & usability by team members

Communication Protocol

- ➔ WiFi ➔ Bluetooth: Portability demands LAN independence

CSM Software

- ➔ Sequential ➔ Concurrent ➔ Parallel**: Listen to Bluetooth while processing frames & improving inter-frame speed
- ➔ Golang ➔ Python: Perception requires Python, multi-language platform creates yet another link exposed to failure



Integration – Virus Edition

Integration Testing Limitations

- ➔ Only one team member had access to a Jetson
- ➔ Jetson needed to run Perception code
- ➔ Heavy documentation on shared code for efficient development

Hardware Tested on Arduino

- ➔ Motor tests run using Arduino C
- ➔ Unable to test Python MotorMan

Bluetooth Communications

- ➔ Replaced Jetson Bluetooth capabilities with OSX Bluetooth Framework for testing

	Assigned	Progress
▾ Mechanical & Circuitry		0%
CAD Mech Parts	Ismael Mercier	0%
CAD Integration slack	Ismael Mercier	0%
Refine CAD models	Ismael Mercier	0%
3D Print & Assemble Mechanical Parts	Ismael Mercier	0%
Integrate motors into 3D printed hull	Ismael Mercier	0%
Battery Management System	Ismael Mercier	0%
▾ Embedded Control		0%
Motor Drivers	Ismael Mercier	0%
▾ Computer Vision		0%
Research CV models/pipeline	Diego Martinez	0%
Object Detection	Diego Martinez	0%
Object Tracking	Diego Martinez	0%
Reading frames in concurrently	Diego Martinez	0%
Optimize pipeline	Diego Martinez	0%
▾ System Control		0%
Blueprint all interfaces & interactions	Diego Martinez, Ike Kilin	0%
Learn iOS App Framework	Diego Martinez, Ike Kilin	0%
Preliminary CSM skeleton	Ike Kilinc	0%
CSM Benchmarking	Ike Kilinc	0%
Refactoring FINAL CSM	Ike Kilinc	0%
User Interface Design	Ike Kilinc	0%
Remote Interface	Ike Kilinc	0%
Bluetooth communication	Ike Kilinc	0%
▾ Management		0%
Create Design Doc	Diego Martinez, Ike Kilin	0%
Spring Break	Diego Martinez, Ike Kilin	0%
Demo #1 Slack	Diego Martinez, Ike Kilin	0%
Integrate App Control & HW System	Diego Martinez, Ike Kilin	0%
Integrate CSM & Perception	Diego Martinez, Ike Kilin	0%
Demo #2 Slack	Diego Martinez, Ike Kilin	0%
Final Presentation	Diego Martinez, Ike Kilin	0%
Final Video & Report	Diego Martinez, Ike Kilin	0%