That's so Fetch

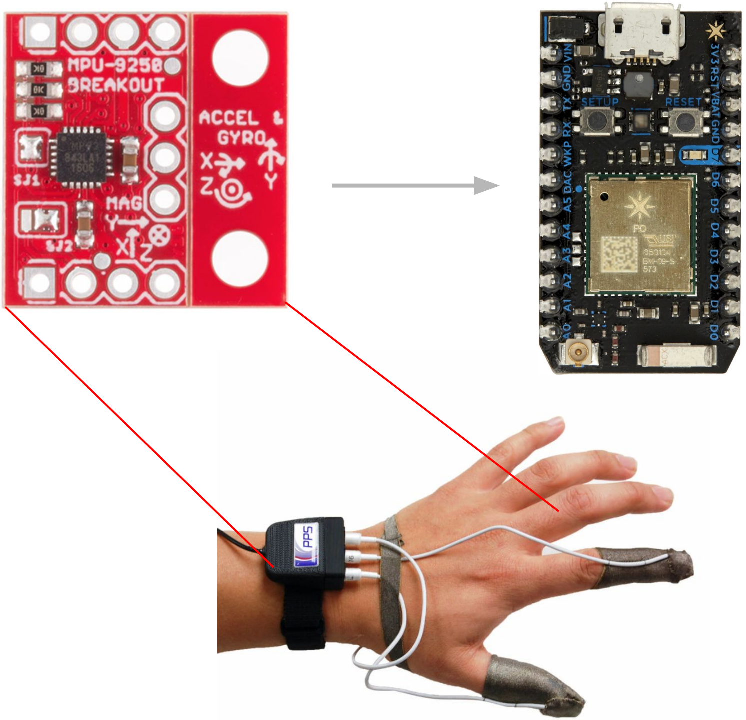
Team B4 (Luca Amblard, Daniel Barychev, Hana Frluckaj)

Application Area

- Motorized Device
  - Anticipates user’s throw using motion sensors on user’s hand
  - Moves to projected landing location
  - Catches ball
  - Returns it to the user
- Geared towards customers who:
  - Are allergic to dogs and want to simulate Fetch
  - Want to throw balls for practice

Solution Approach (User)

- IMUs: MPU-9250
  - 3 axis Accelerometers
  - 3 axis Gyroscopes
- One IMU placed on user's knuckles and another IMU placed above user's wrist
- Trajectory prediction from data produced on pre-throw
  - Velocity
  - Angle
- IMUs wired to Particle Photon
- Photon intakes data and will transmit to robot's Jetson



Solution Approach (Dog)

- Data from Photon sent over Wifi to Jetson on motorized base
- Jetson runs Kalman filters that get ball position/angle upon release
  - Computes estimated movement needed to anticipate final position
- Jetson controls motors through PID system
- Once caught, base brings ball back to user’s position
- Starting position is always 2m in the direction of the prethrow

Software Design

Kalman Filter

IMU Acceleration (X'', Y'', Z'')

Prediction

Project the state ahead

$$x_{k+1} = Ax_k + Bu_k$$

Project the error covariance ahead

$$P_{k+1} = AP_kA^T + Q$$

Correction

Compute the Kalman Gain

$$K_k = P_kH^T(HP_kH^T + R)^{-1}$$

Update the estimate via measurement

$$x_k = x_k + K_k(z_k - Hx_k)$$

Update the error covariance

$$P_k = (I - K_kH)P_k$$

Initialize R, P, Q once

IMU Position (X, Y, Z)

Ball Position Prediction

$$Horizontal\ distance, x = V_x t$$

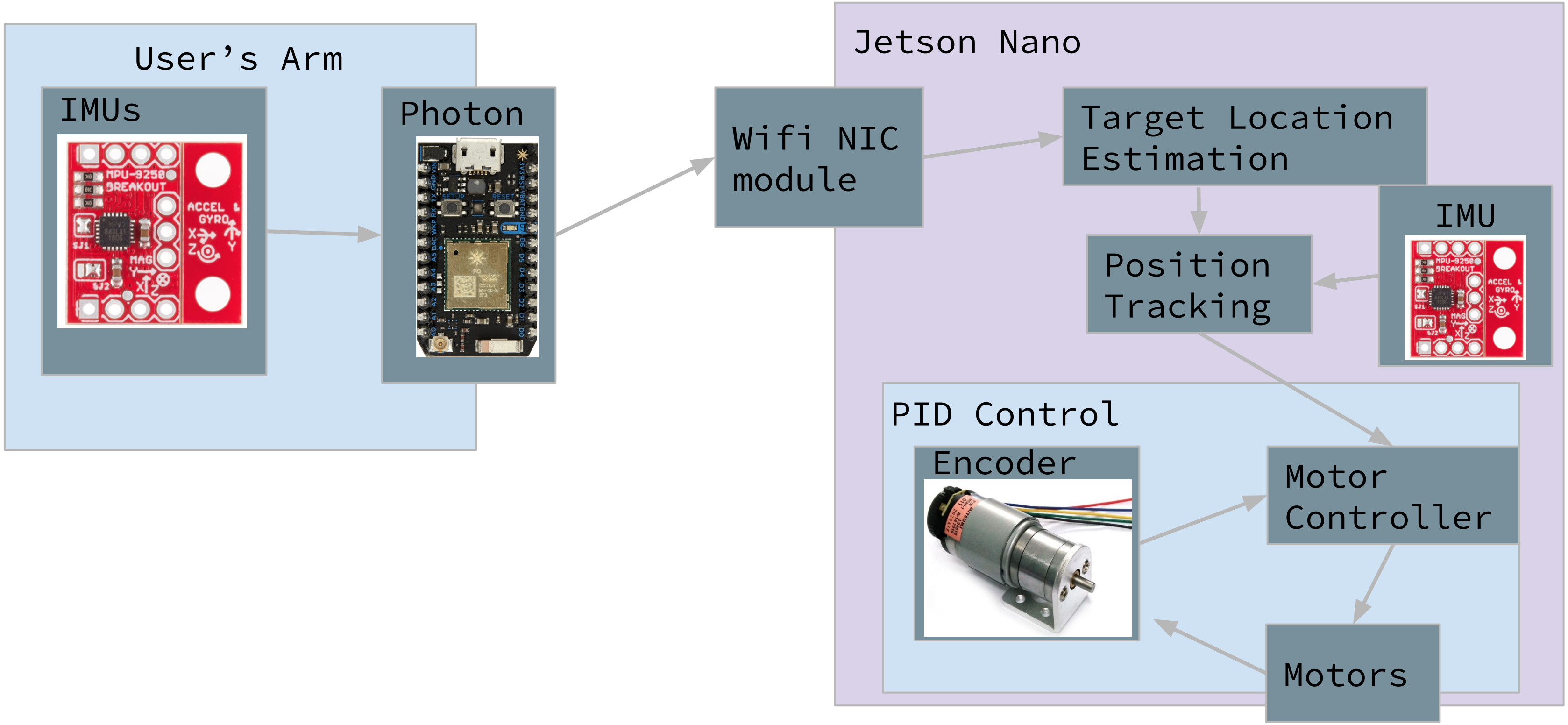
$$Horizontal\ Velocity, V_x = V_{x_o}$$

$$Vertical\ distance, y = V_{yo}t - \frac{1}{2}gt^2$$

$$Vertical\ Velocity, V_y = V_{yo} - gt$$

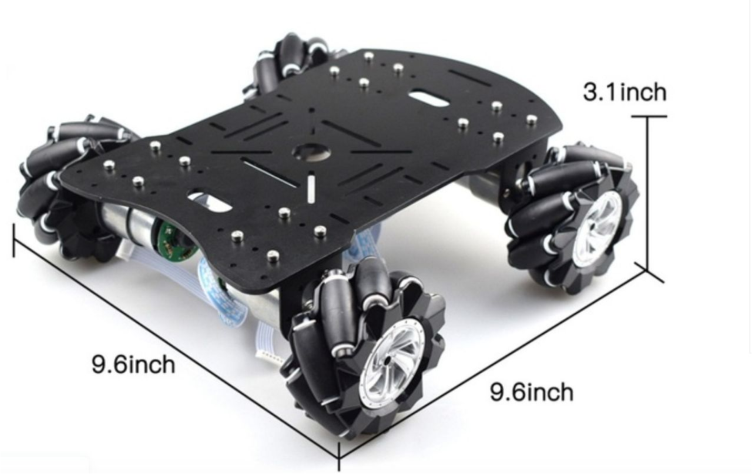
Estimated Ball Position (X, Y, Z)

Hardware Block Diagram



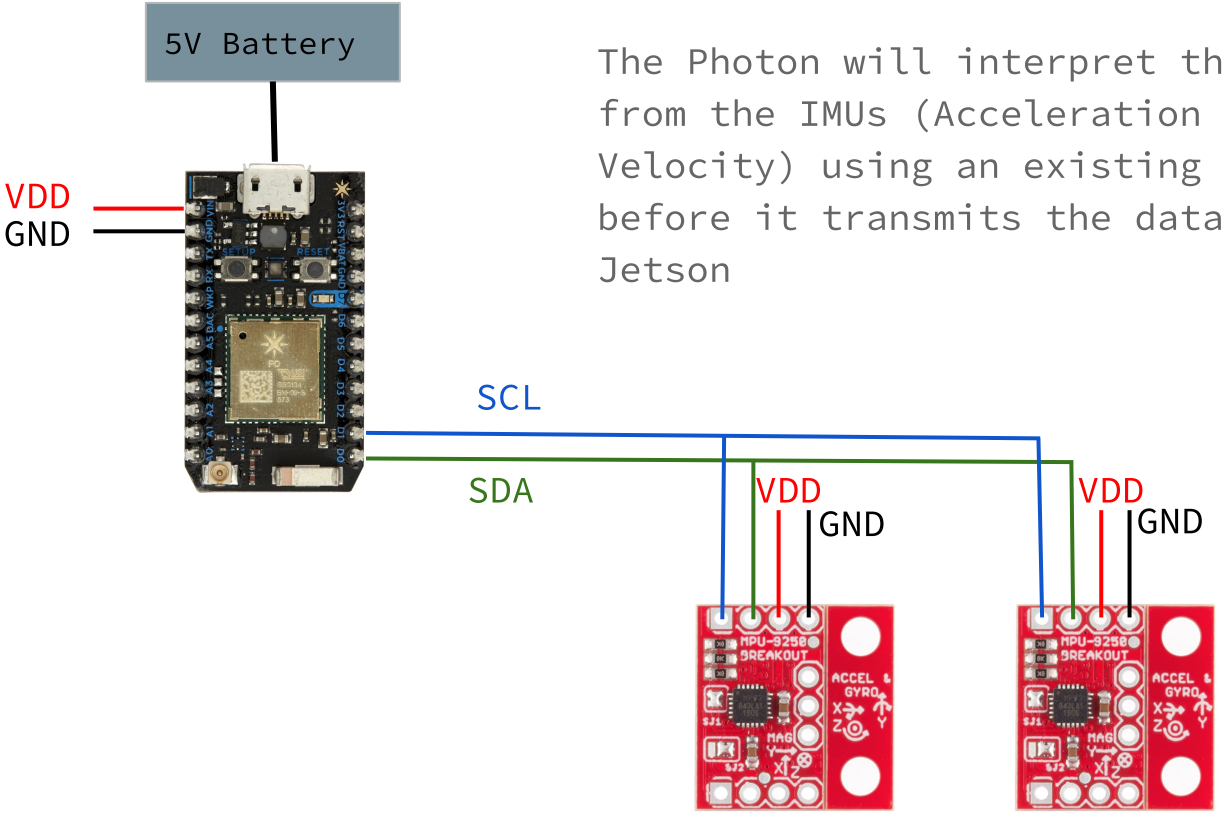
Hardware Details

- Moebius Metal Mecanum Omni Wheel Robot Car Kit
  - Four 12v 330 rpm omnidirectional motors
  - Motor encoder outputs 360 pulses per each rotation cycle



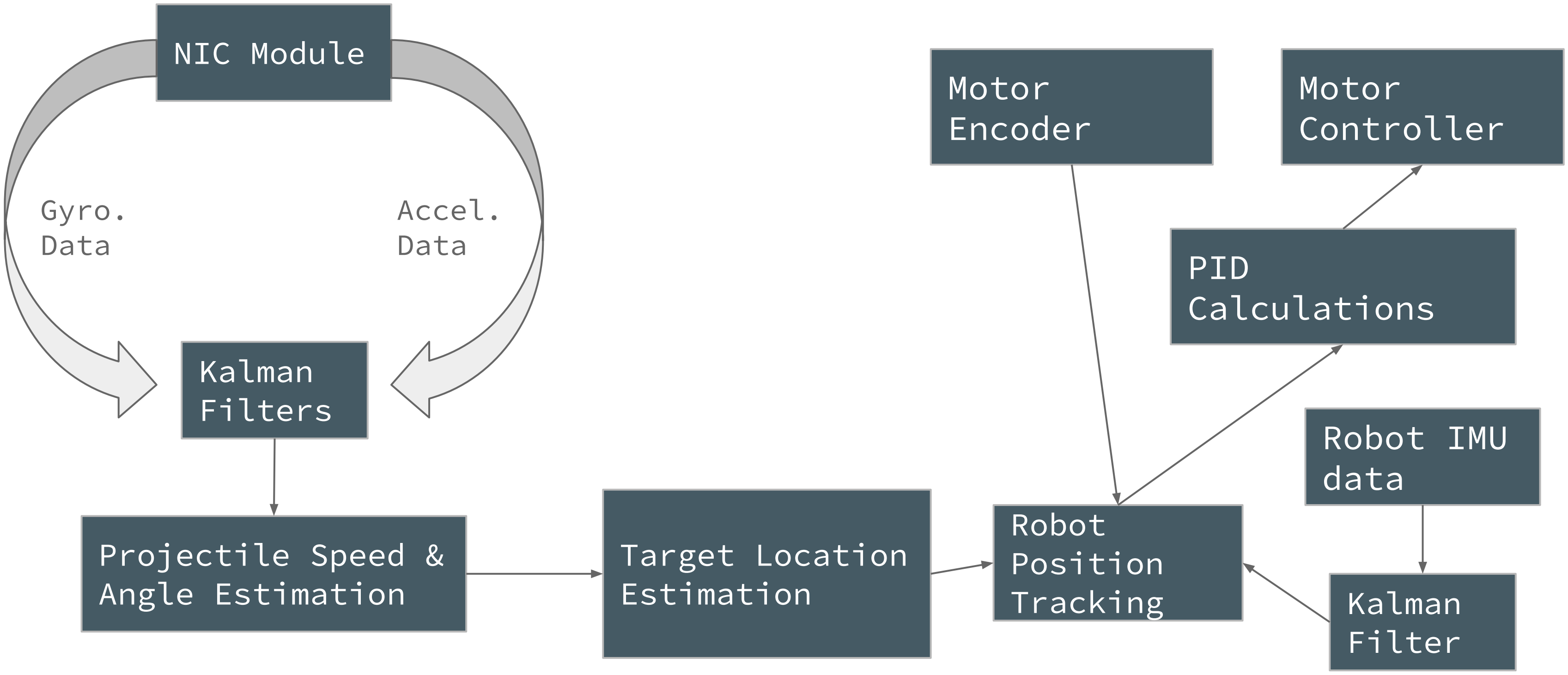
Note: The majority of lateral movement will take place during the pre throw. The goal of omnidirectional motors is to always have the robot face the user

Implementation Plan: Photon-IMU i2c Interface



The Photon will interpret the I2C data from the IMUs (Acceleration and Angular Velocity) using an existing library before it transmits the data to the Jetson

Implementation: Jetson Nano Computation



Metrics / Requirements

Process	**Specs**
Success Rate *(#balls thrown v. #balls caught)*	> 50%
User throw range *(distance between user and dog)*	2m radius
Device retrieval range	1m radius
Device basket diameter	25cm
Projected *(prethrow)* angle vs. actual angle	< 5%
Estimated ball landing position - actual landing position	< 12.5cm
IMU-Motor communication latency	< 200 ms
Position error after reset	< 12.5cm

Validation

- The pre throw accurately predicts the thrown angle so robot can perform minimal lateral movement for the actual throw
  - 5% allowed error measured by lateral variation of the prethrow predicted release position from actual release prediction and computing how these angles relate
- Actual landing position within 12.5cm of predicted landing position
  - Measure from the actual landing position v. estimated using a ruler
- When resetting for another throw, the error computed in our position computation should be within 12.5cm for rethrow assuming the same starting points.
  - If this fails, we will implement a physical reset system to determine a new set of throw parameters

Schedule

Tasks:	24 Feb	2 Mar	9 Mar	16 Mar	23 Mar	30 Mar	6 Apr	13 Apr	20 Apr	27 Apr	4 May
Motion Sensing											
Analyze process and planning	DONE										
Ordering parts	DONE										
Reading data through I2C											
IMU data transmission (wifi)											
IMU-position mapping											
Dog Simulator											
Analyze process and planning	DONE										
Projectile prediction - physics											
Order parts for Dog	DONE										
IMU data transmission (wifi)											
Build mobile base for dog											
Controls mobile base (wheels and motors)											
Catching/retrieval			BREAK								
Returning the ball											
Integration & Testing								CARNIVAL			
Slack Time											
	Key:										
		Daniel	Luca	Hana	All						