The University of Texas at Austin
Aerospace Engineering
and Engineering Mechanics
Cockrell School of Engineering

November 2022

DESIGN REVIEW #1

USRC - Simulation / Estimation

Izaac Facundo
Joseph Flores
Nicholas Franken
Neel Pandey
Preston Thomas
William Wang

PROJECT OVERVIEW

OUR PROJECT

- The overall need of this project is to develop drones capable of sensing and avoiding obstacles with a minimized risk of collision or damage to property in GPS-denied environments.
- Obtain a system that is able to take real time inputs from actively flying drones and output a state estimation with three dimensional error ellipsoid visualization
- Collision avoidance concerning other drones and environmental obstacles
- The application of this system would be for the increase of commercial drone delivery operations

PROJECT OVERVIEW

- Using an Extended Kalman Filter (EKF) to turn error into ellipsoid visualization
- Physical data taken from the sensors on board the drone will be used in tandem with a powerful simulation through Gazebo
- The collision avoidance system will make sure that no two error ellipsoids around their respective drones will intersect

ESTIMATION & SIMULATION

TEAM STRUCTURE

- Meeting Times:
  - Monday 2-3 PM
  - Friday 2-5 PM
  - More time if needed
- Informal Split of responsibilities
  - Nick is person of contact
  - Expertise distributed among us

GOALS

- Implement an extended Kalman filter (EKF) on drone positions
- Develop flight software to control drones precisely
- Visualize drones and drone behavior using simulation software
- Develop drones' ability to successfully avoid obstacles
- Develop architecture for communication between drones
- Design system for software-software and software-hardware integration

PROFESSIONAL RESPONSIBILITY

- Collision Avoidance: Need a nearly guaranteed success rate of avoiding obstacles, inanimate and animate alike, to minimize injury and/or damage to property
- This affects certain aspects of our drone operation:
  - Maximum velocity
  - Altitude
  - Avoidance trajectories
- More factors will be considered in the future
  - Acoustics
  - Drone end of life

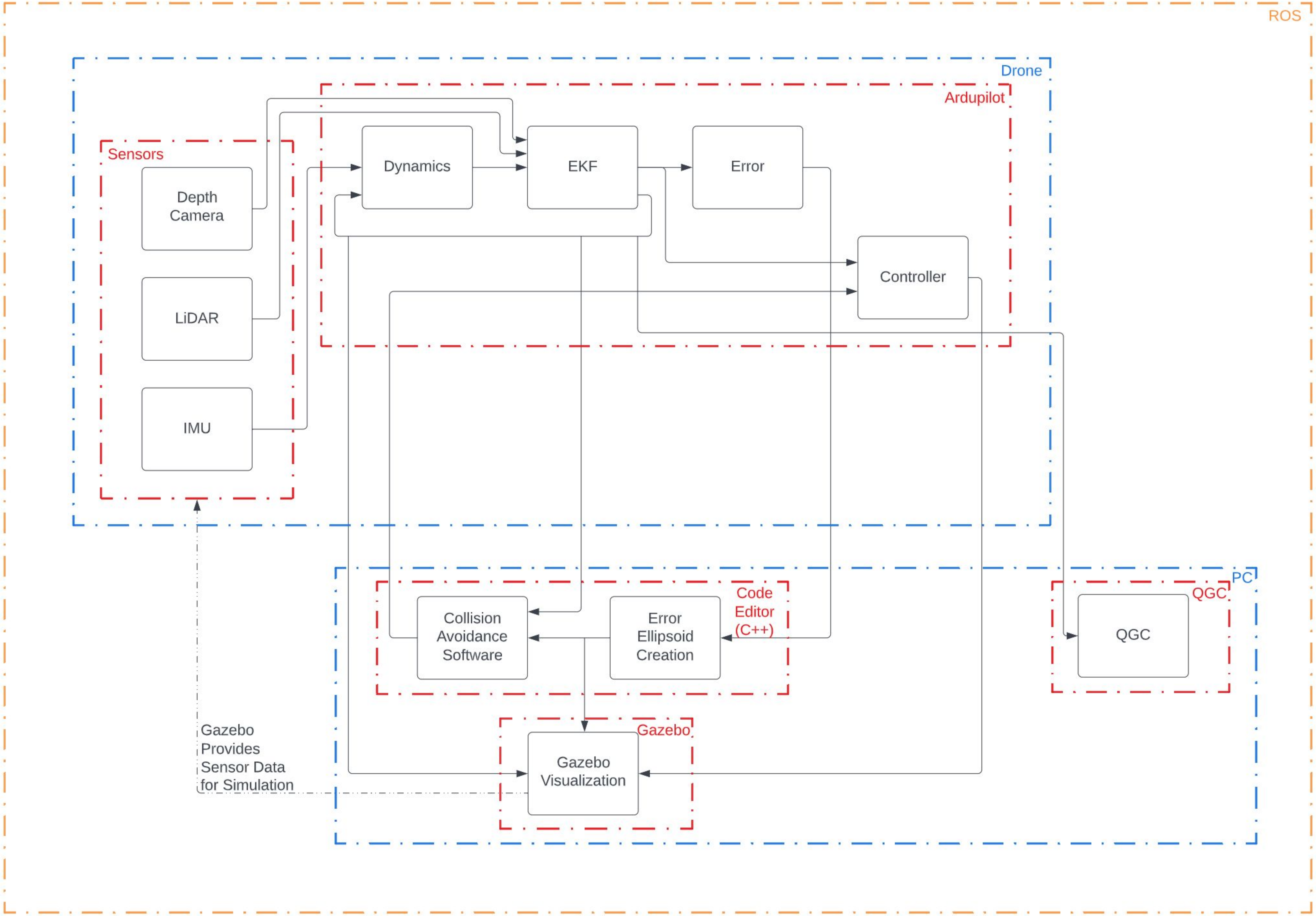
SOFTWARE & BLOCK DIAGRAM

OVERALL OPERATING SYSTEM (OS)

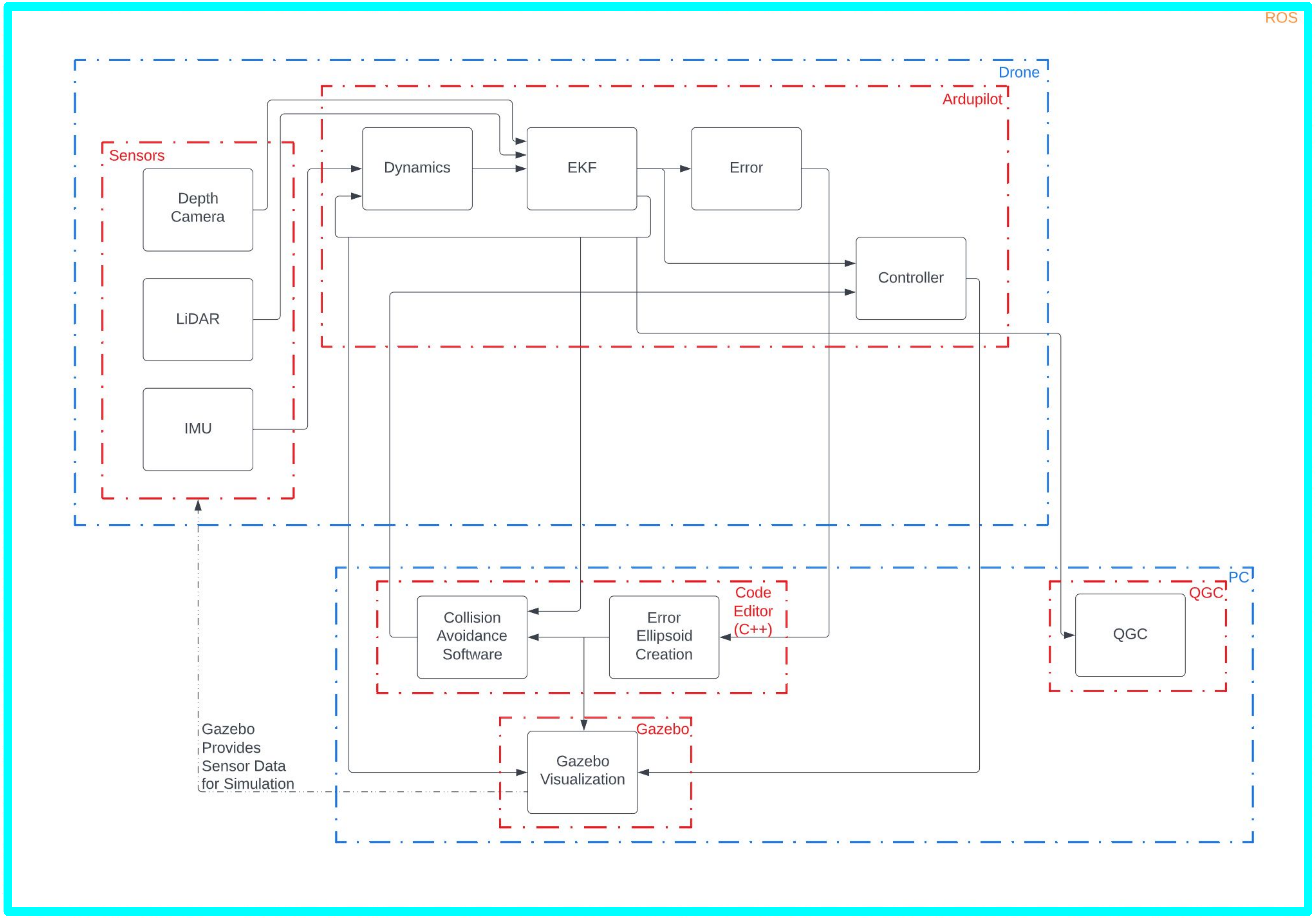
- For developmental purposes, we will be using Linux/Ubuntu. This will allow for more freedom and control when developing software. Certain software like the Robot Operating System (ROS) are developed specifically for operation on a Linux OS.

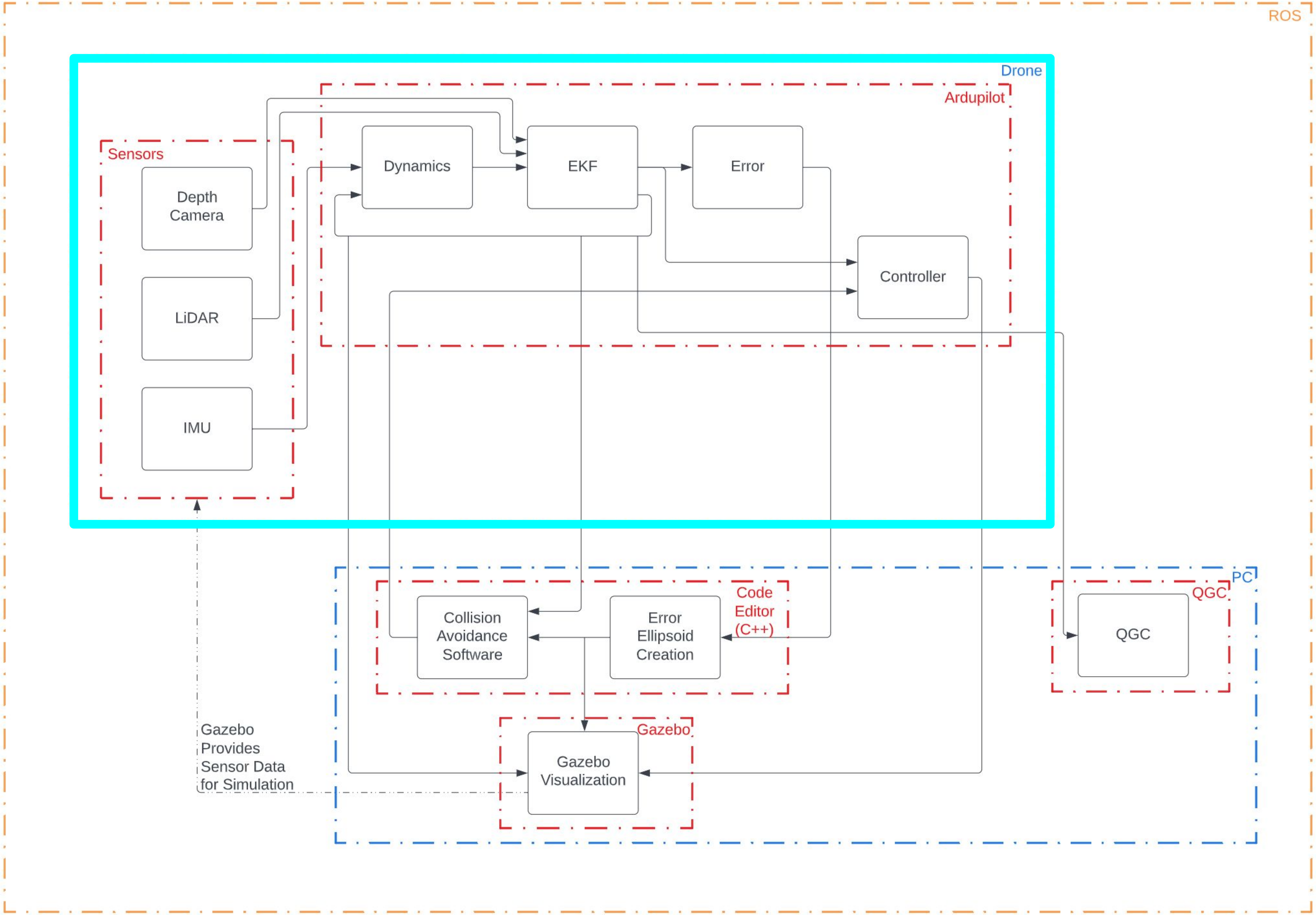
SOFTWARE LIST

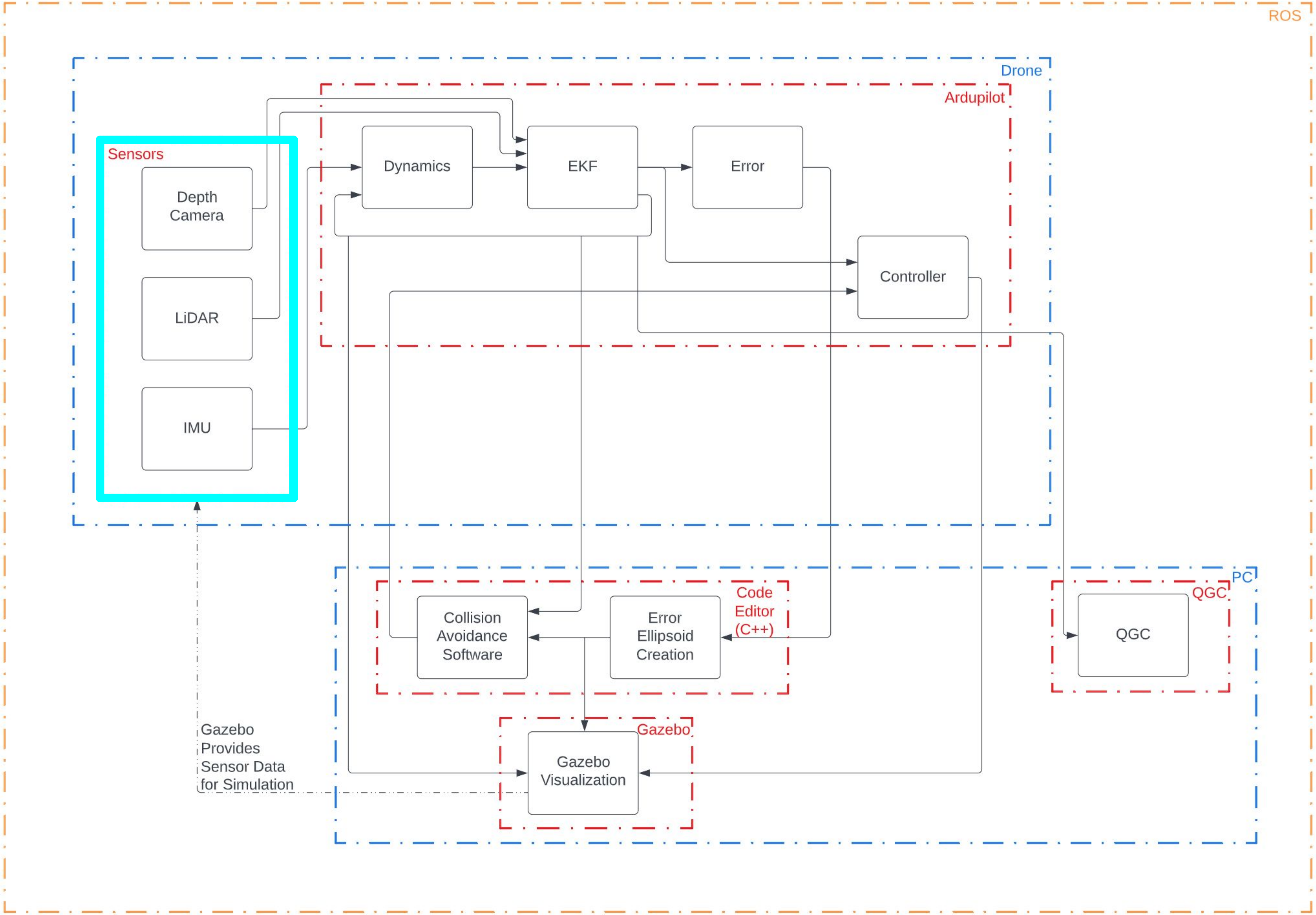
- ArduPilot
- Code Editor (MATLAB,VSCode, Visual Studio, Eclipse, etc.)
- Gazebo/Simulink
- QGroundControl (QGC)
- Robot Operating System (ROS)
- Sensor Software (LiDAR, Depth Camera, etc.)



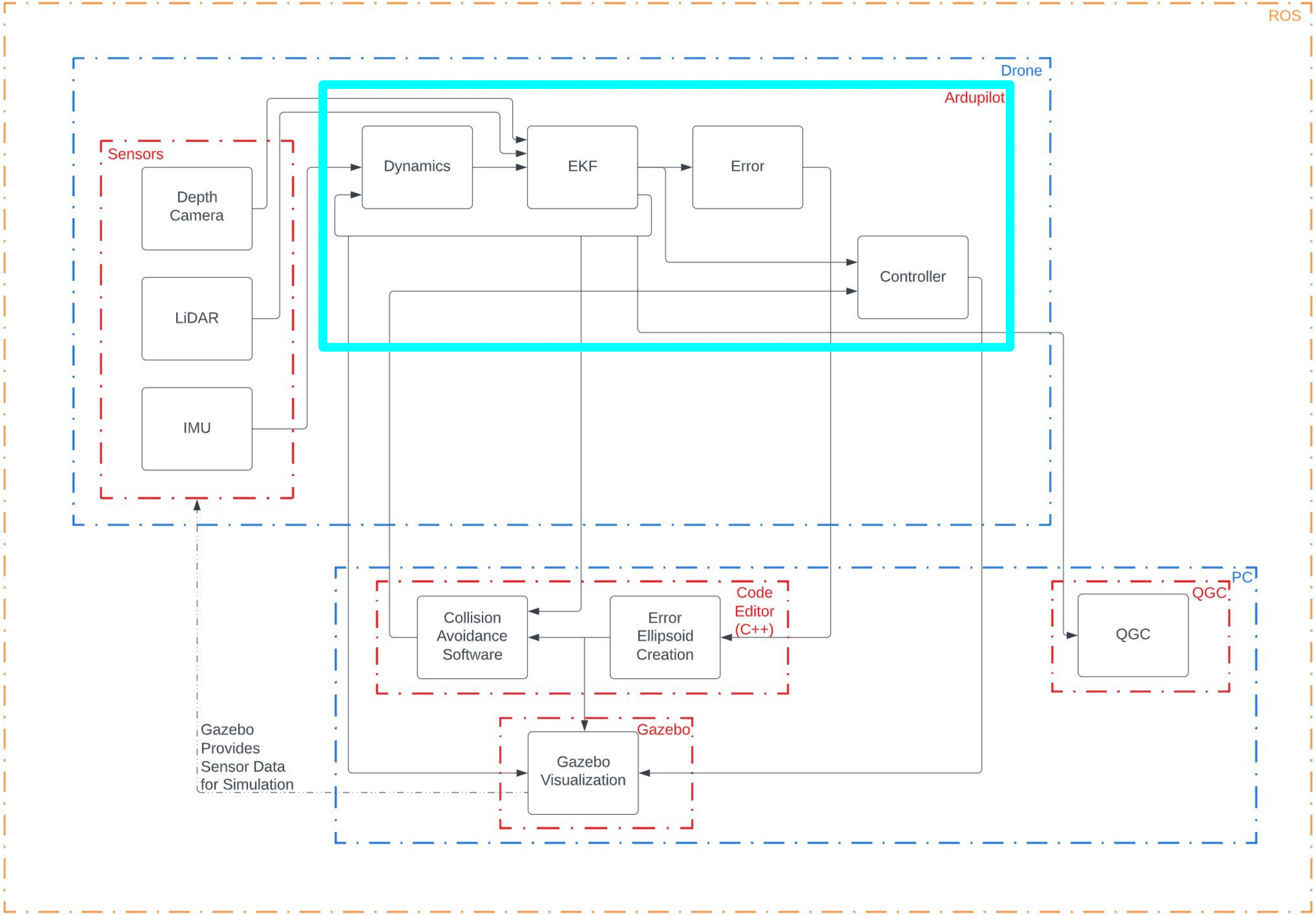
ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation


ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation

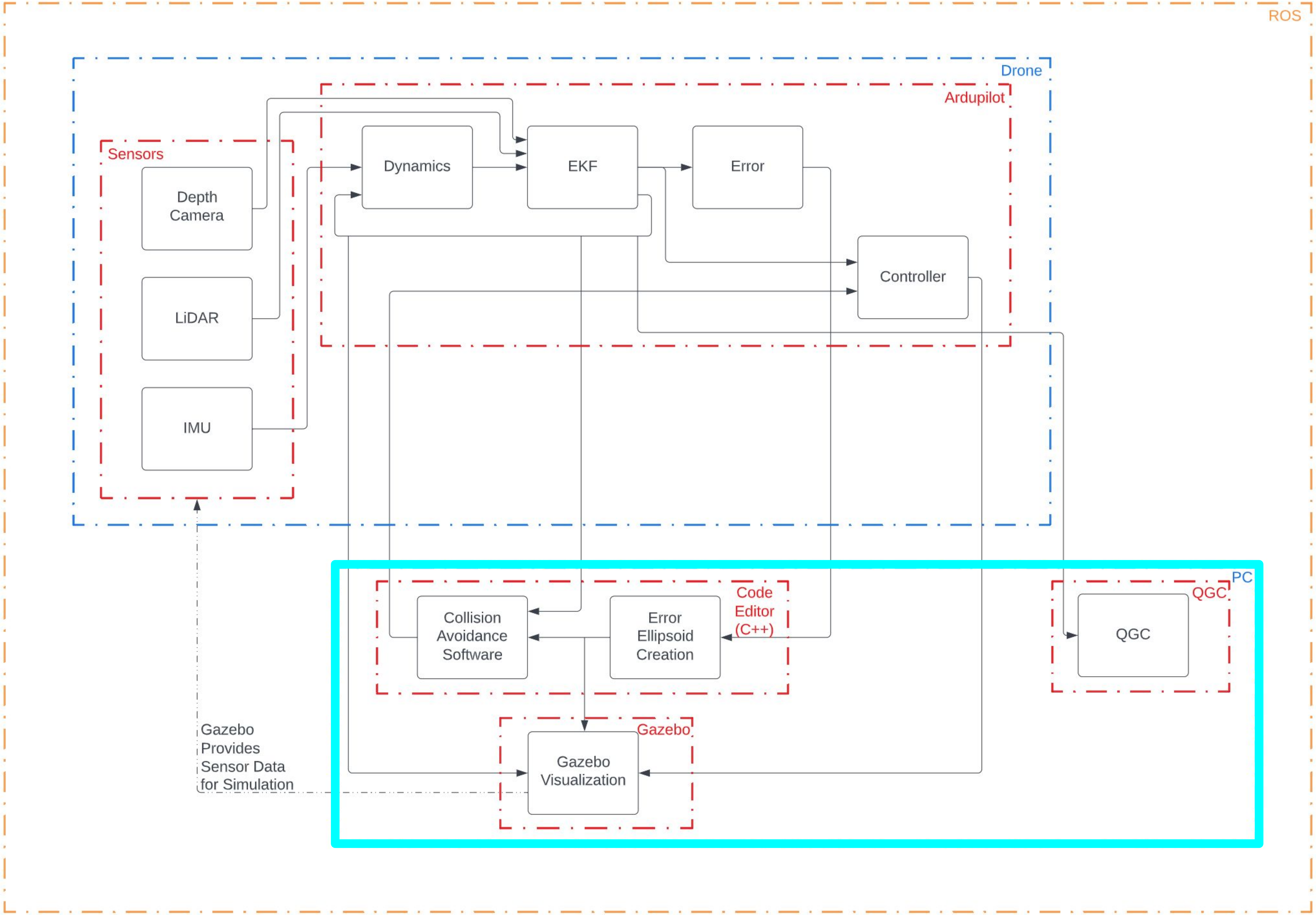

ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation



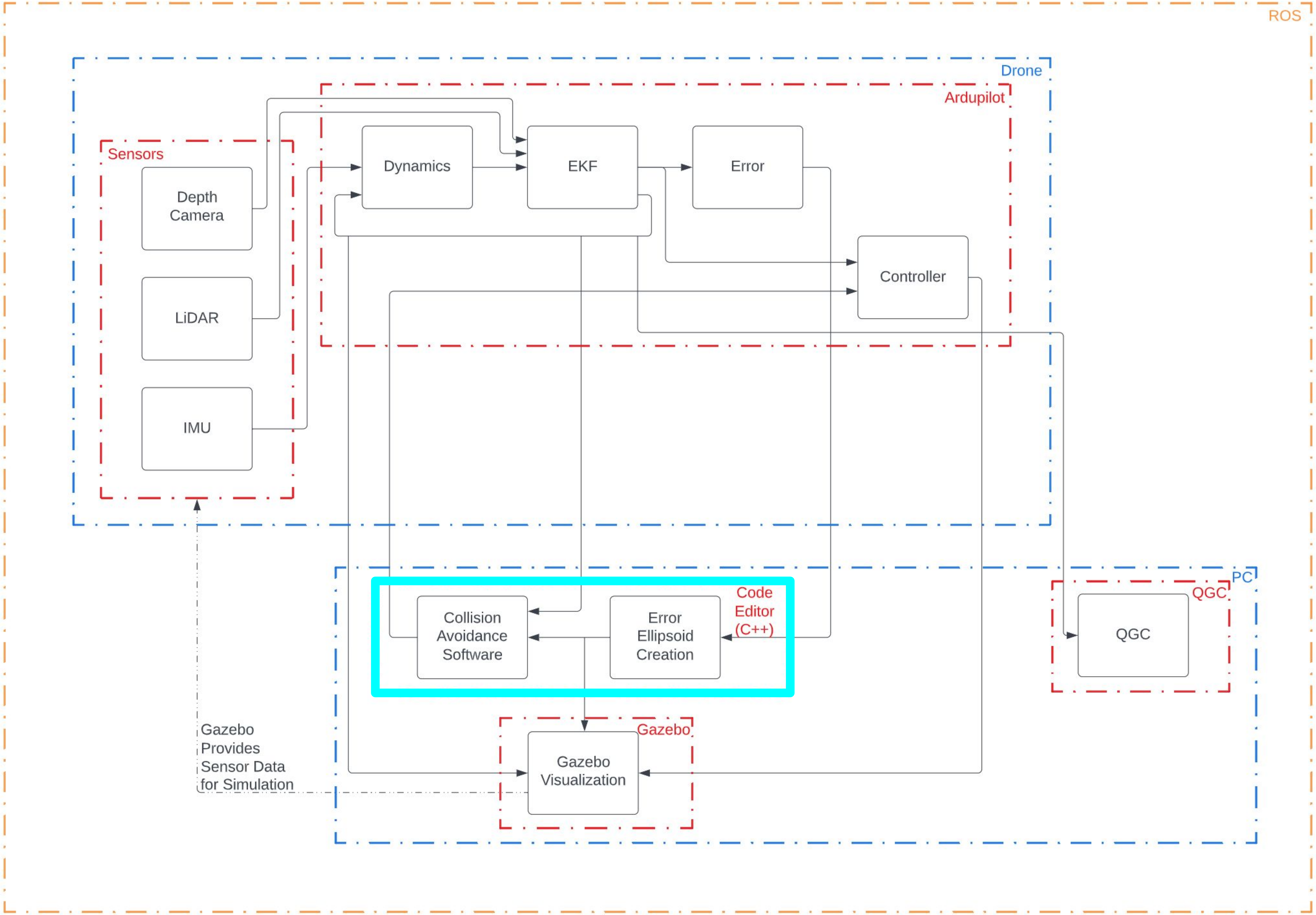
ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation

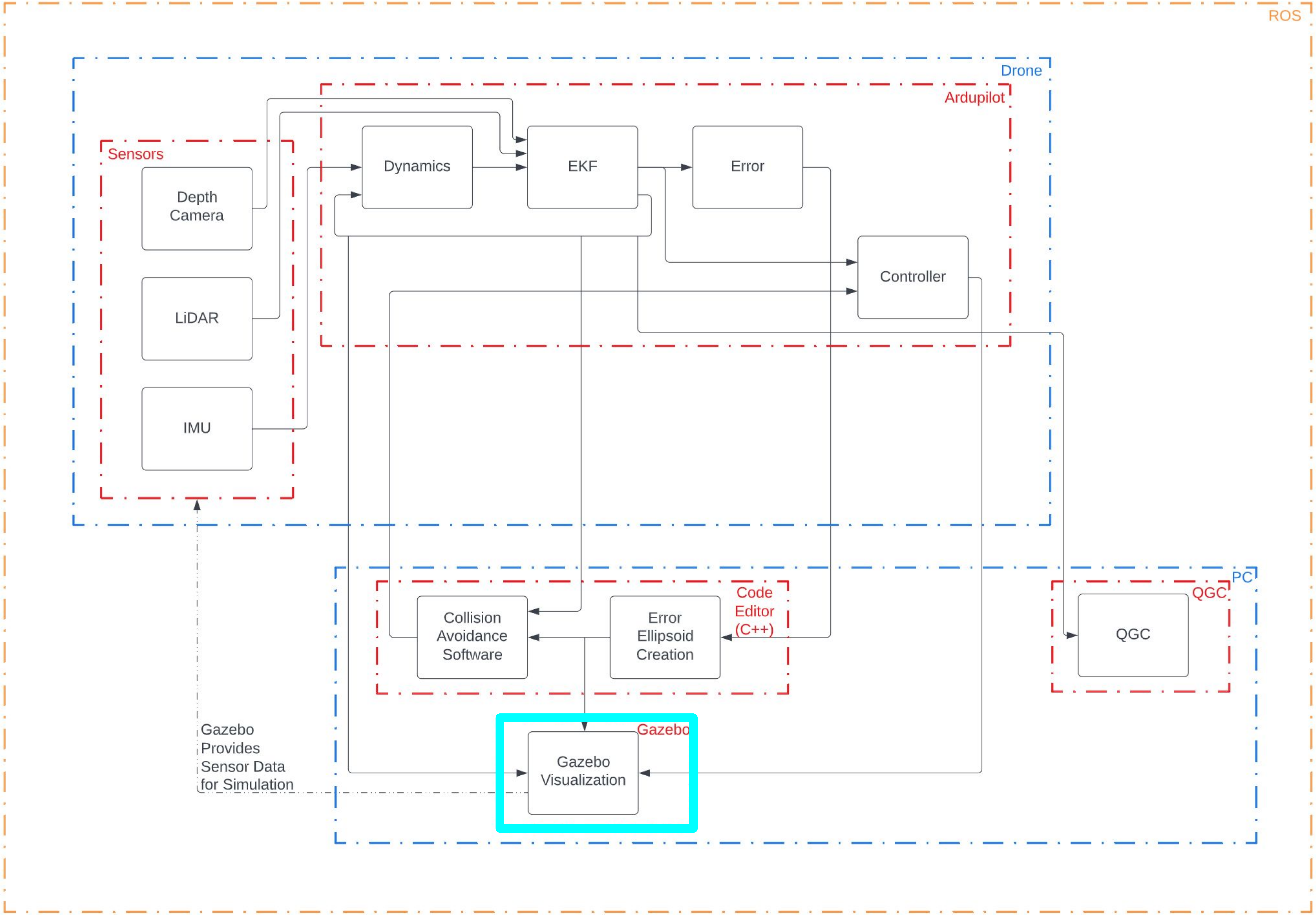


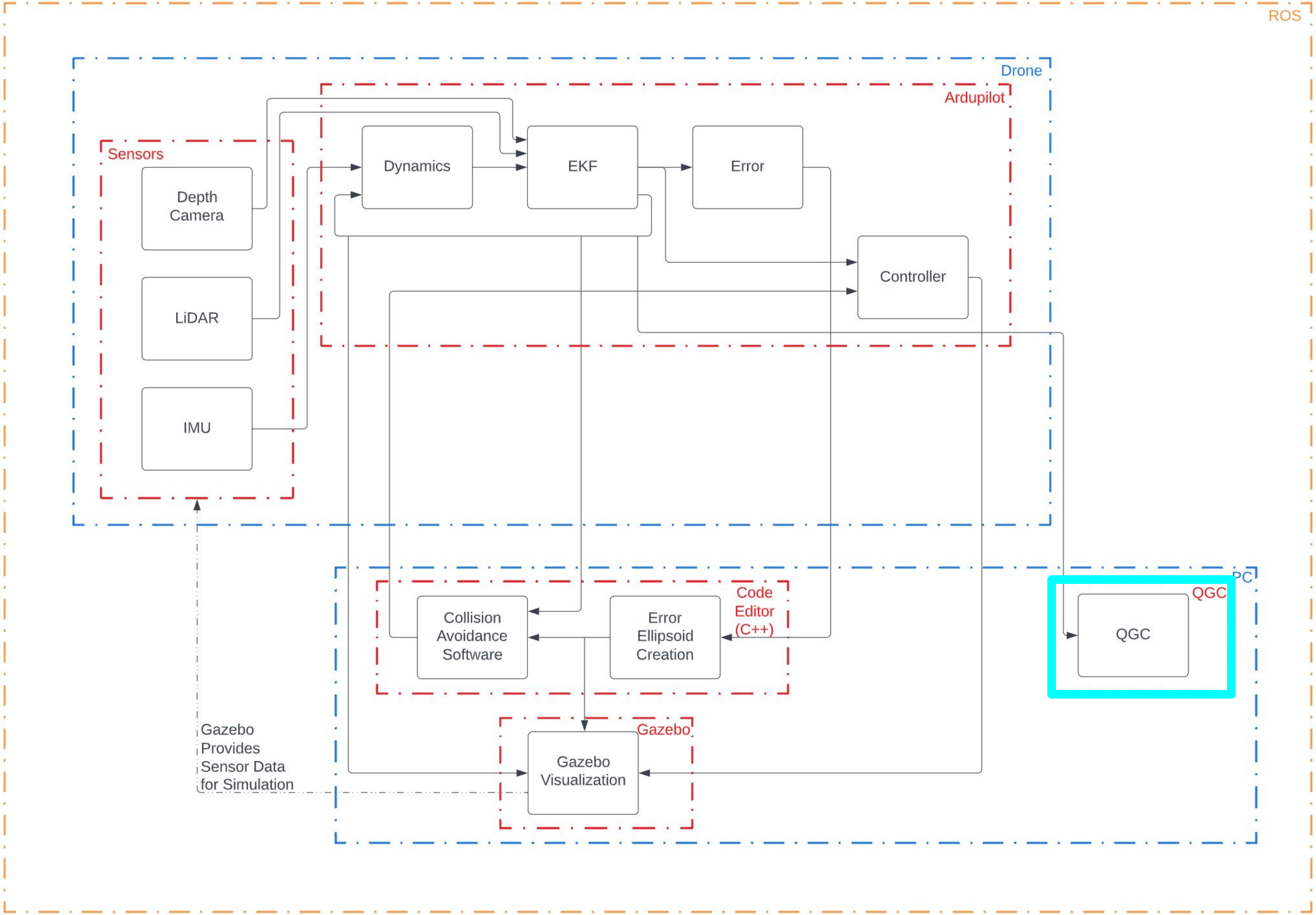
ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation



ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation


ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
QGC
QGC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation

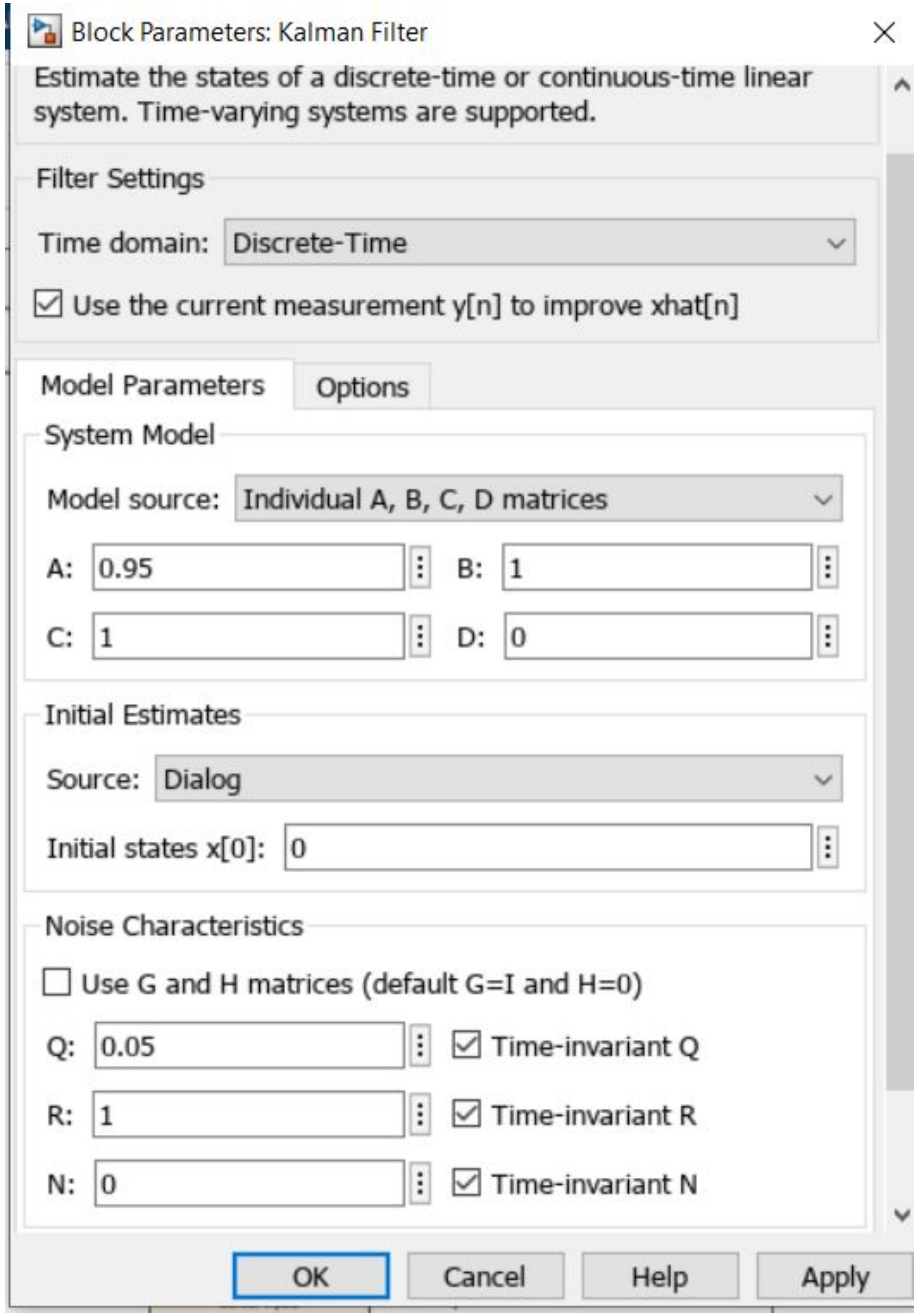

ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation

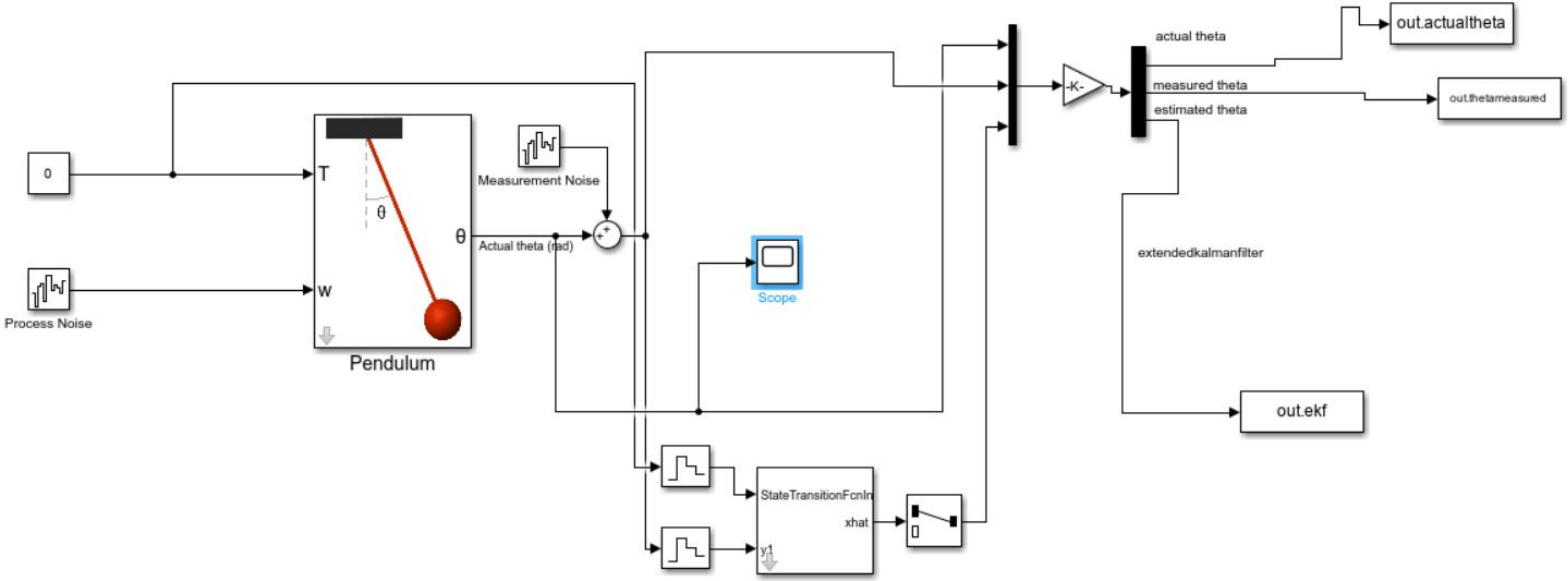

ROS
Drone
Ardupilot
Sensors
Depth Camera
LiDAR
IMU
Dynamics
EKF
Error
Controller
PC
Code Editor (C++)
Collision Avoidance Software
Error Ellipsoid Creation
QGC
QGC
Gazebo
Gazebo Visualization
Gazebo Provides Sensor Data for Simulation

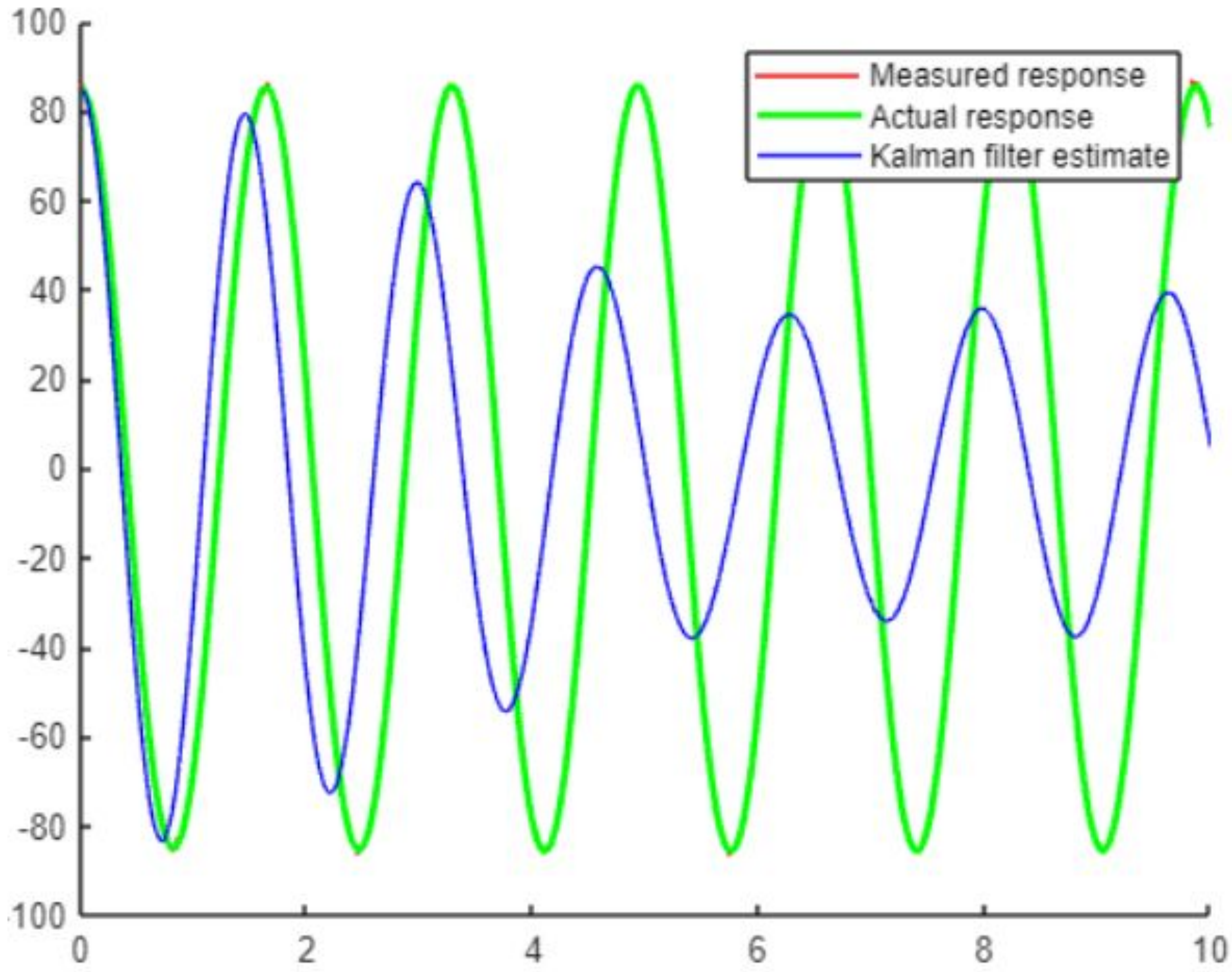
PENDULUM SIMULATION

Extended Kalman Filter (EKF)

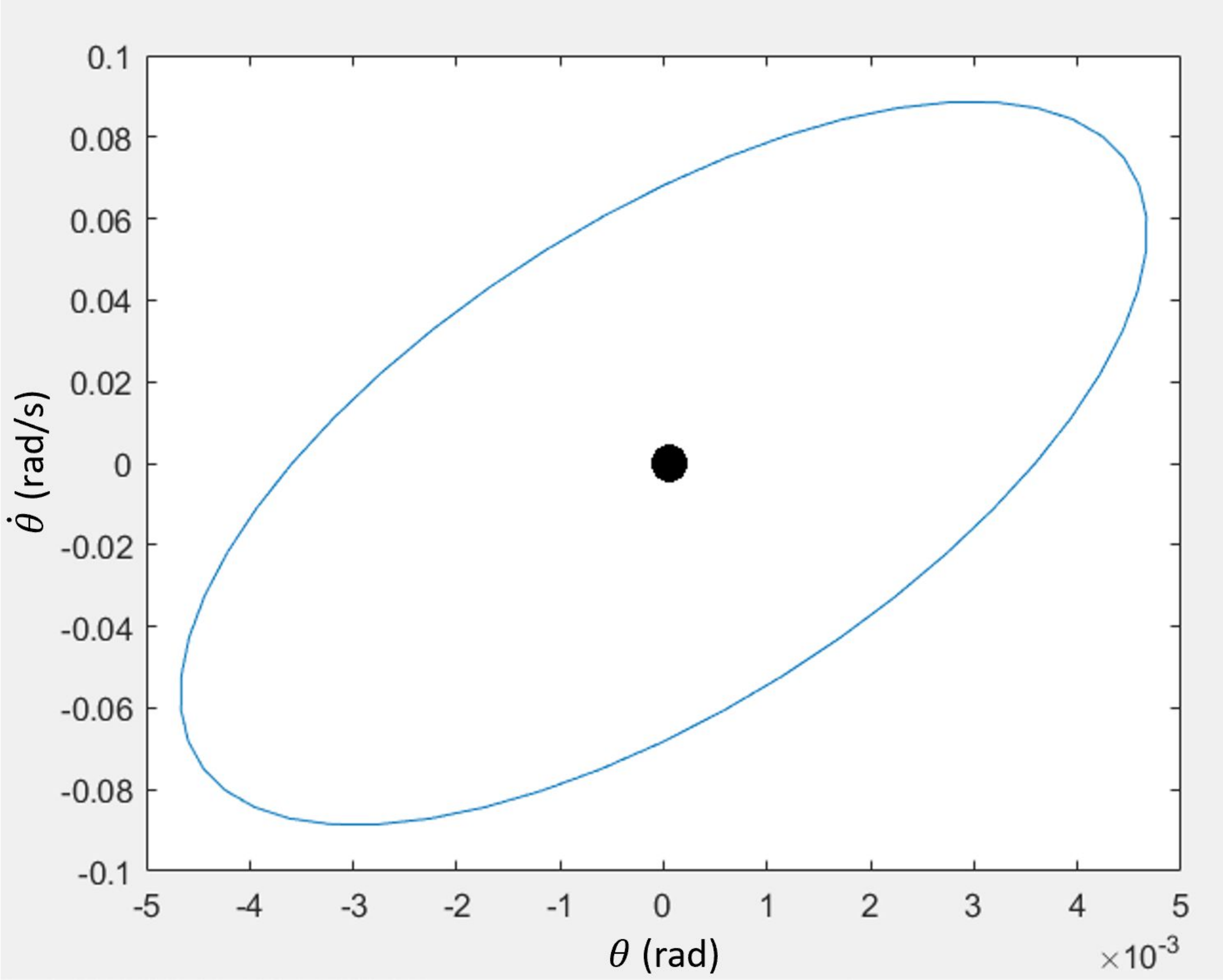
- A Kalman Filter is a filter that takes a less than perfect dynamic model and noisy measurements to provide very accurate state estimations for a system. This filter only works for linear systems.
- Two covariance matrices: Q and R (process and measurement noise respectively)
- An EKF can be used in non-linear systems but requires more computation. The general concept is the same, and the Q and R covariance matrices are still very important.




0
Process Noise
T
w
θ
θ
Pendulum
Measurement Noise
Actual theta (rad)
Scope
StateTransitionFcnIn
xhat
y1
-K-
actual theta
measured theta
estimated theta
out.actualtheta
out.thetameasured
extendedkalmanfilter
out.ekf


Measured response
Actual response
Kalman filter estimate
100
80
60
40
20
0
-20
-40
-60
-80
-100
0
2
4
6
8
10

Error Ellipsoid Visualization



- Error ellipsoid constructed from the final 2 x 2 covariance matrix of angle and angular rate
- Covariance matrix remained constant after the initial change, meaning the error ellipsoid visualization was steady after a short period
- From ellipsoid, the difference between maximum and minimum possible angular position is 0.5°, meaning that predicted angular position had uncertainty of ± 0.25°.

POTENTIAL FIELD SIMULATION

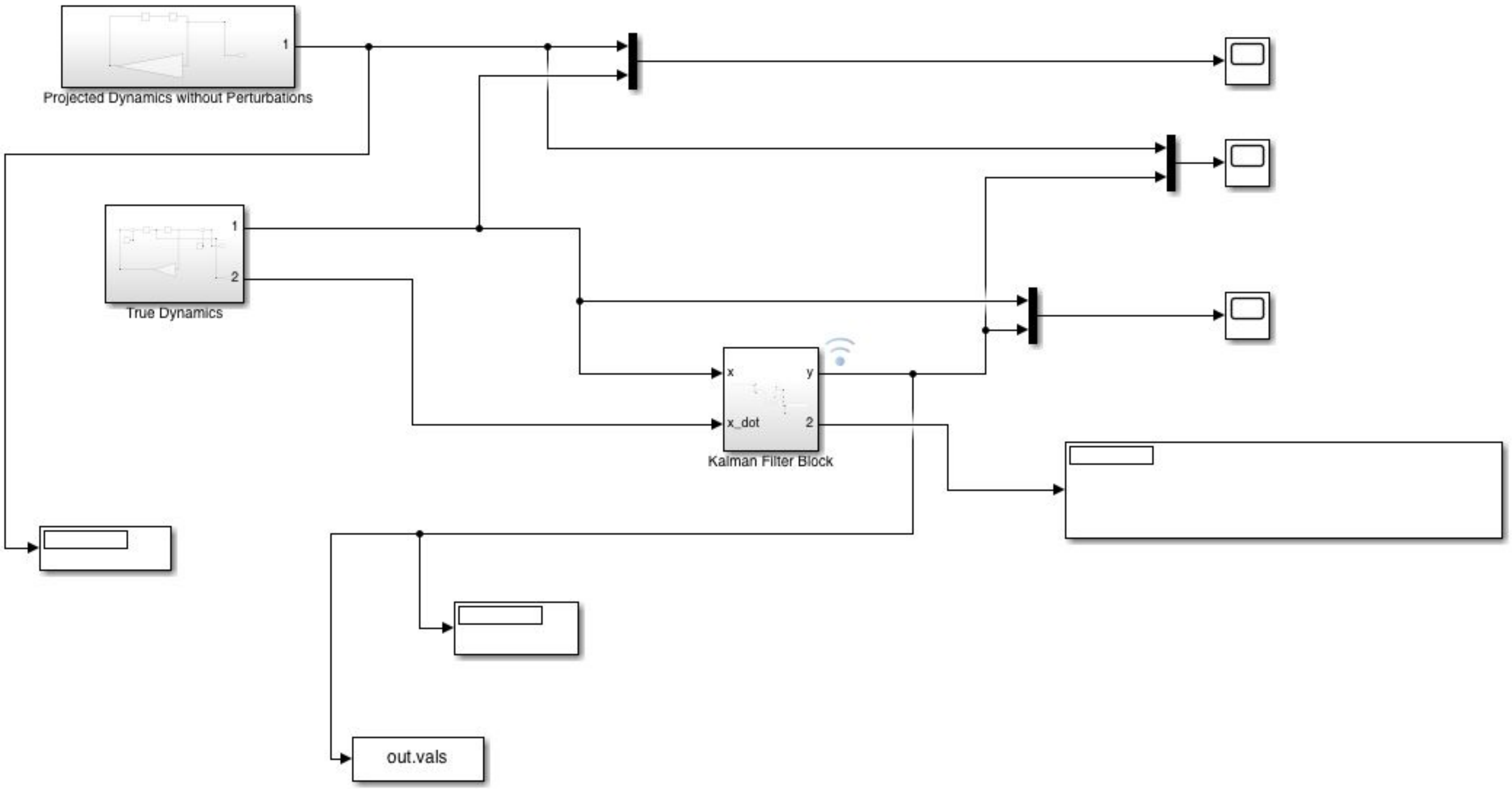
Potential Field Dynamics:

- The main motivation in creating a potential field model was to have a two dimensional model that a Kalman filter could be applied to.

- The potential field was designed to impart an acceleration in the x direction and the y direction based on the location of the particle in the X-Y plane.

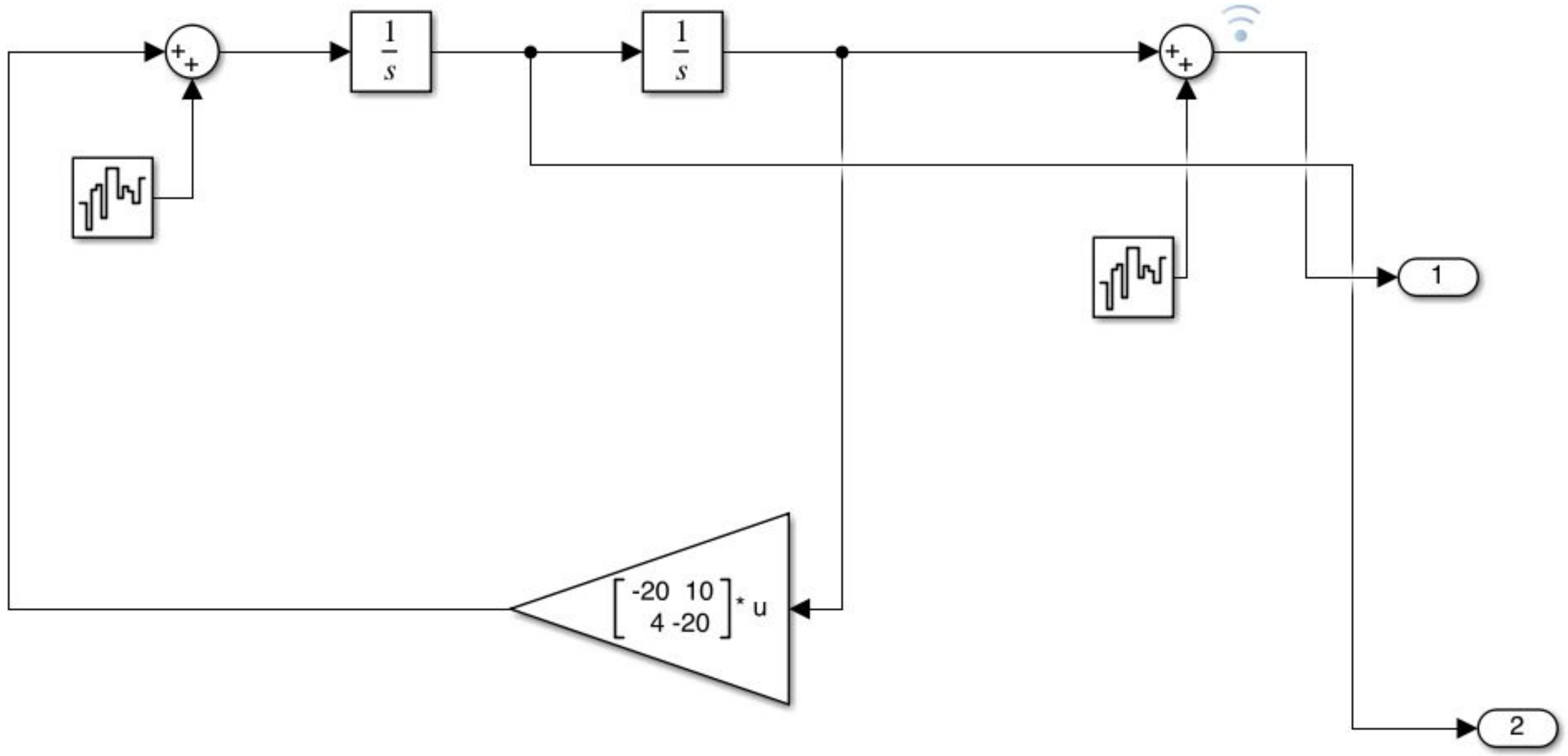
- The accelerations were chosen to be:
  X_Acceleration= -20x+10y
  Y_Acceleration= 4x+ -20y
  This created a coupled system.

Potential Field Block Diagram:



Potential Field Block Diagram:

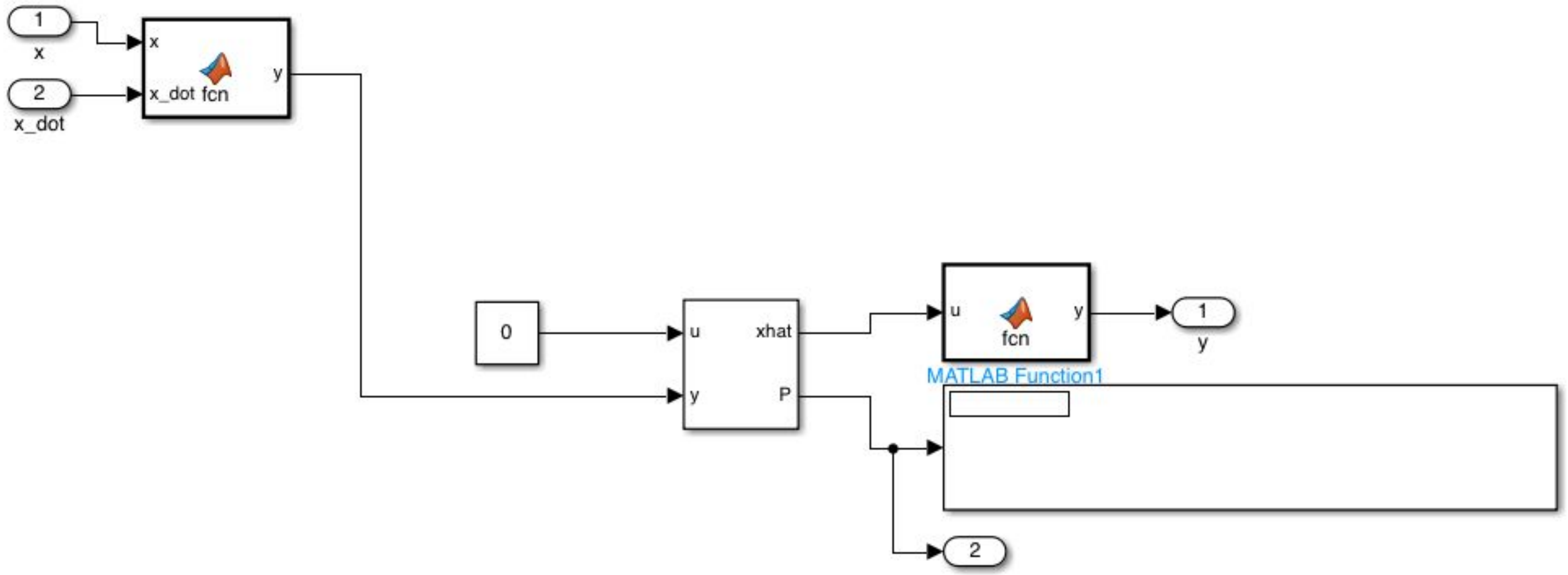
True Dynamics Block:



- The true dynamics were calculated by integrating the differential equations and adding measurement and process noise (chosen at this stage to give a noisy output that still approximately matched the output of the dynamics without noise)

Potential Field Block Diagram:

Kalman Filter Block:



- The Kalman filter requires an input of the projected dynamics within the block. The input was done in state space model form.
- The Kalman filter also requires the input for the actual dynamics found on the previous slide.
- Kalman filter outputs a covariance matrix, which in this case is 4 x 4 because there are 4 states: x, x_dot, y, y_dot

Potential Field Covariance and Error Ellipse:

- The noise is kept constant so the covariance matrix is constant as well
- The values for the covariance matrix that applied to the x and y error were isolated and extracted to create a 2 x 2 matrix. This matrix was then used in a MATLAB script to find the error ellipse.
- Using this 2 x 2 matrix, the error ellipse could be created by implementing an algorithm created for this purpose [1]. This is done by finding the eigenvalues of the matrix and square rooting them, then multiplying by a constant of proportionality to get the major and minor axes.
- Angle of the error ellipse is determined by finding the eigenvectors and taking a ratio and the inverse tangent
- The values of the state for x and y were also outputted to MATLAB for use in the script.
- The error ellipse was then plotted for every point by using the x and y of the state as the center. The visualization was done via a for loop and the drawnow functionality and the state was also plotted.

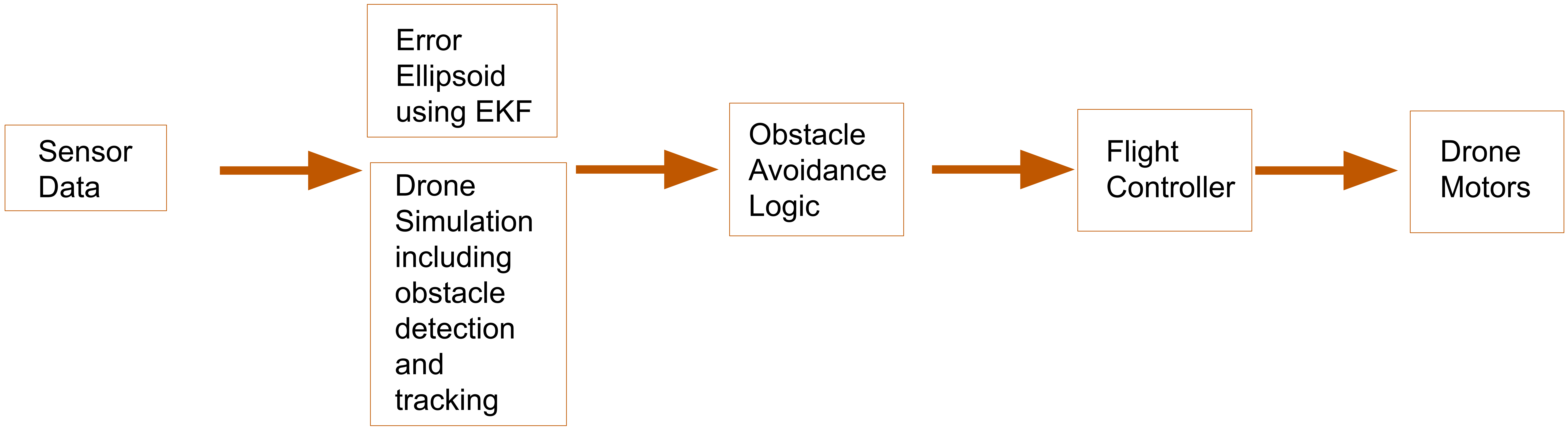
VISUALIZATION

NEXT STEPS

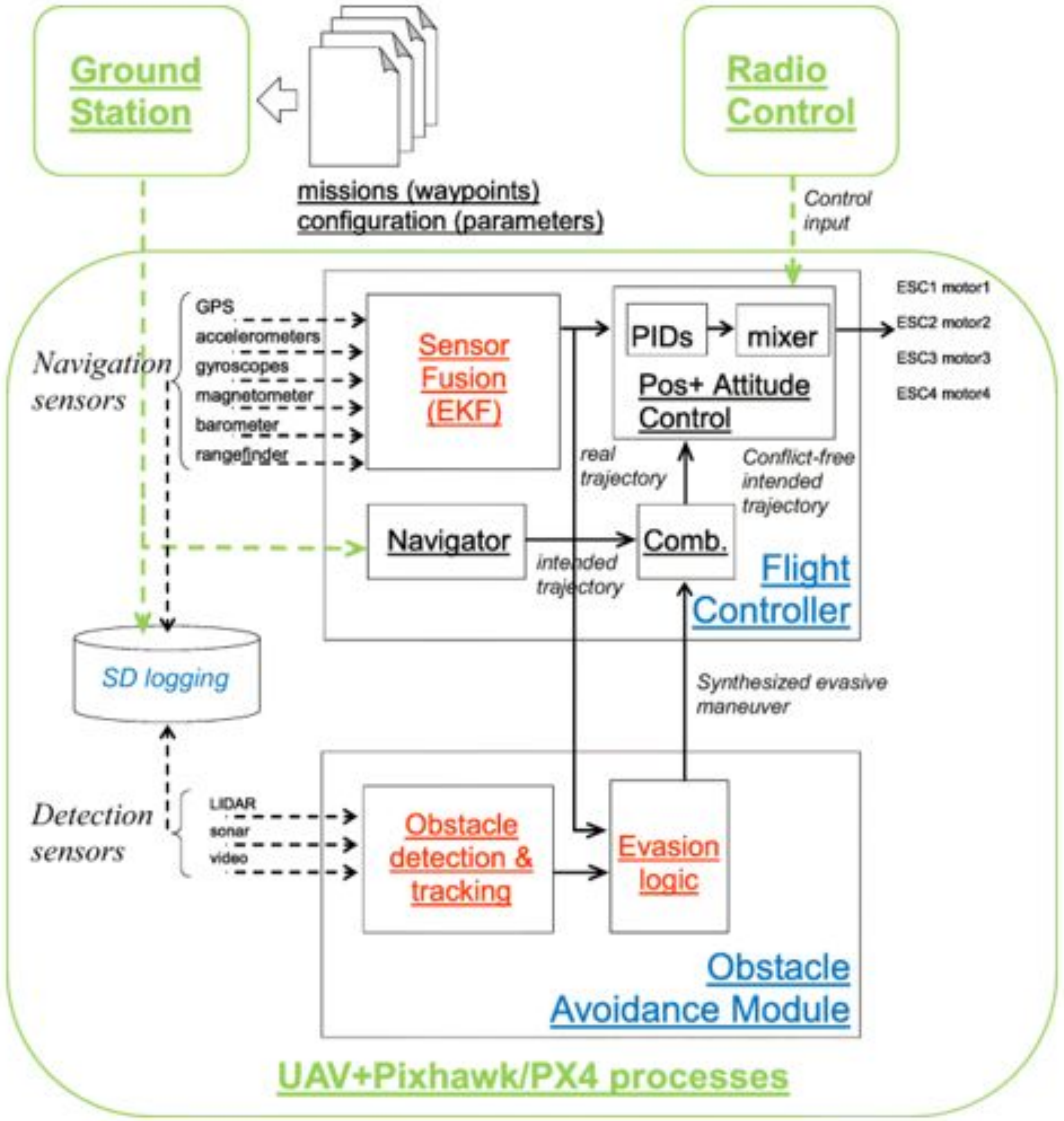
Future Work

- Finalize and polish the code for error ellipse creation in the 2D model
- Research how to apply the extended kalman filter to a three dimensional system
- Deciding how to further implement an EKF for our main purpose

Drone System Pipeline



Fully Integrated Solution



THANK YOU.

USRC - Simulation / Estimation

- Izaac Facundo
- Joseph Flores
- Nicholas Franken
- Neel Pandey
- Preston Thomas
- William Wang

The University of Texas at Austin
Aerospace Engineering and Engineering Mechanics
Cockrell School of Engineering