MATLAB EXPO

MATLAB과 PX4를 활용한 드론 비행제어 소프트웨어 개발

Development of drone flight control software using MATLAB and PX4

문정호 교수, 청주대학교 무인항공기학과

MathWorks®

CONTENTS

Autonomous System Research Group(ASRG)

- **Autonomous systems from ground to space based on aerospace engineering**
  - Flight dynamics modeling and control law of UAV, UAM/AAM, SLV
  - Robot operating system based robotic embedded systems & simulation
  - AI-based image processing, decision making and estimation (inc. SLAM)
  - System design & flight test of fixed-wing, rotary-wing, and VTOL air vehicles

Speciality of research team

1st Controls for autonomous aircraft

2nd Machine learning for aerospace

3rd Convergence of robotics & aerospace engineering

Prof. Jung-Ho Moon
Ph.D in aerospace engineering
Department of Unmanned Aircraft Systems
- Flight dynamics & simulation
- Flight control law design
- Collision avoidance
- UAV system design & flight test

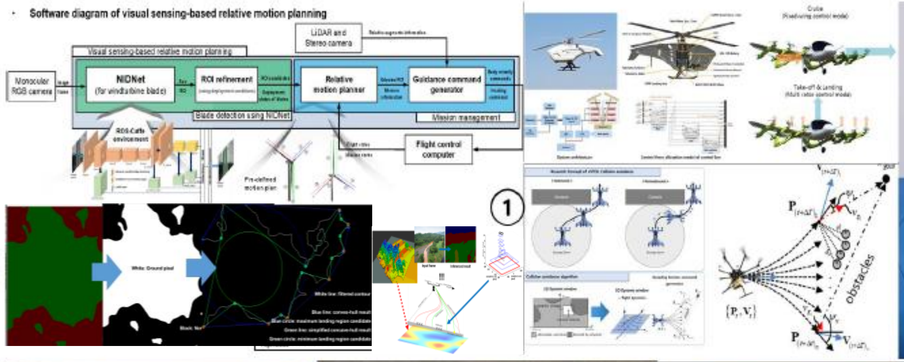
Chief GNC engineer in Koreanair('05~19)

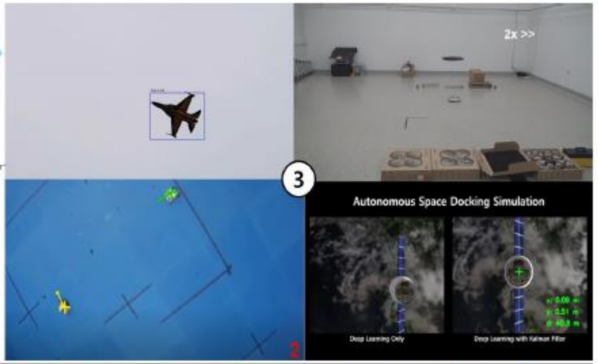
Prof. Sungwook Cho
Ph.D in aerospace engineering
Department of Aeronautical and Mechanical Engineering
- Heterogenous sensor fusion
- Learning-based target recognition and state estimation
- Edge device development for UAVs
- ROS-based unmanned system design

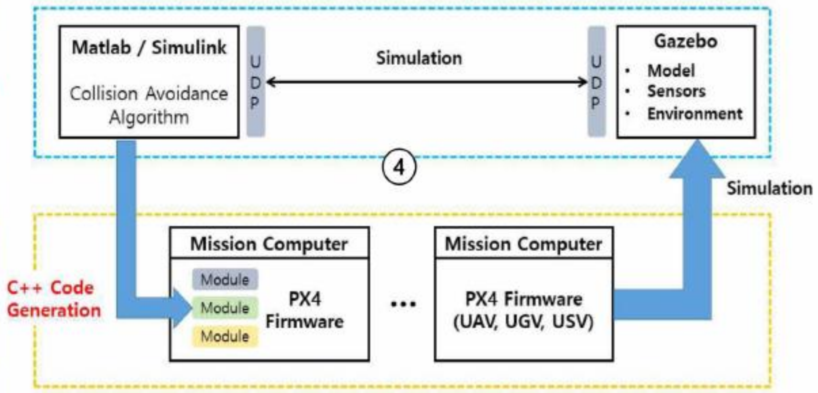
Senior engineer in Nearthlab('17~19)

Prof. Sang-Hyeon Kim
Ph.D in aerospace engineering
Department of Unmanned Aircraft Systems
- Intelligent robotics
- AI based vision system for robot
- S/W for UAV/UAM system
- Intelligent smart factory system

Senior engineer in Samsung Electronics Co.,Ltd('18~20)

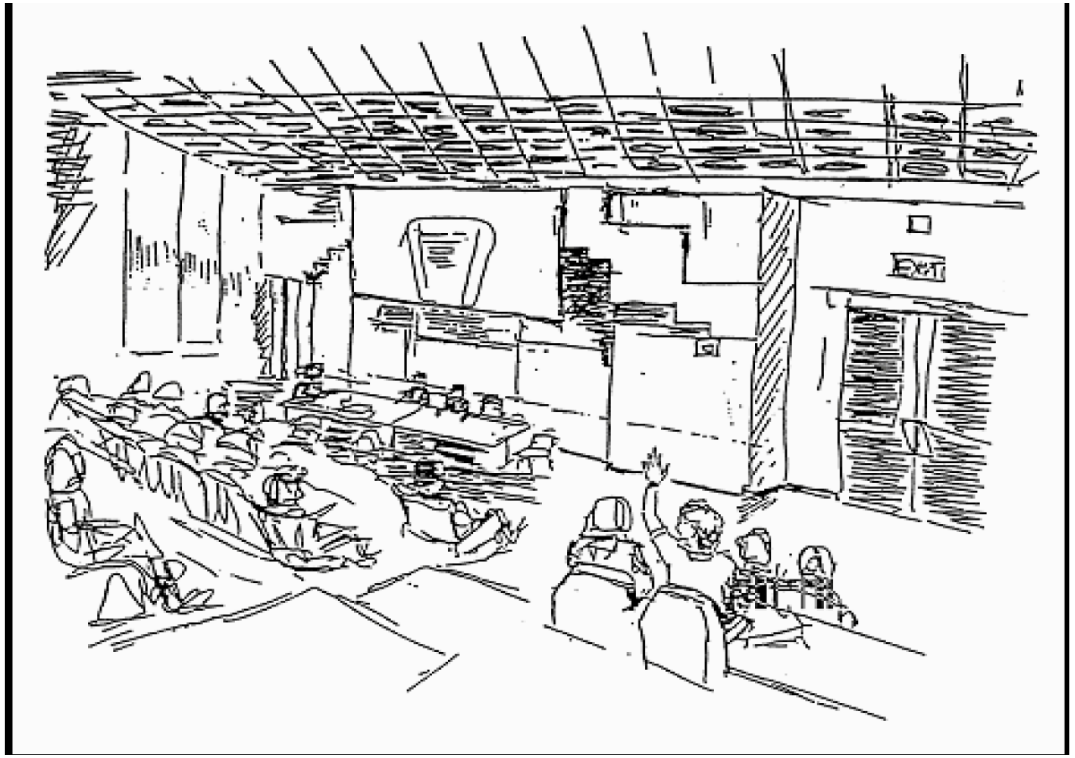


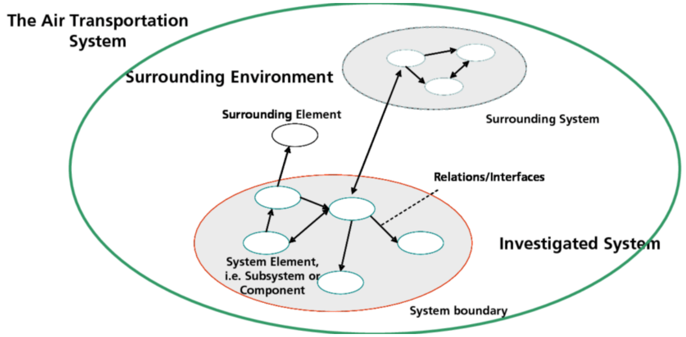




Why does ASRG emphasize aerospace engineering?

- **RTCA, DO-178B Software consideration in airborne systems and equipment certification**
  - Modeling & Simulation, Design Optimization, Human factors, Airworthiness certification





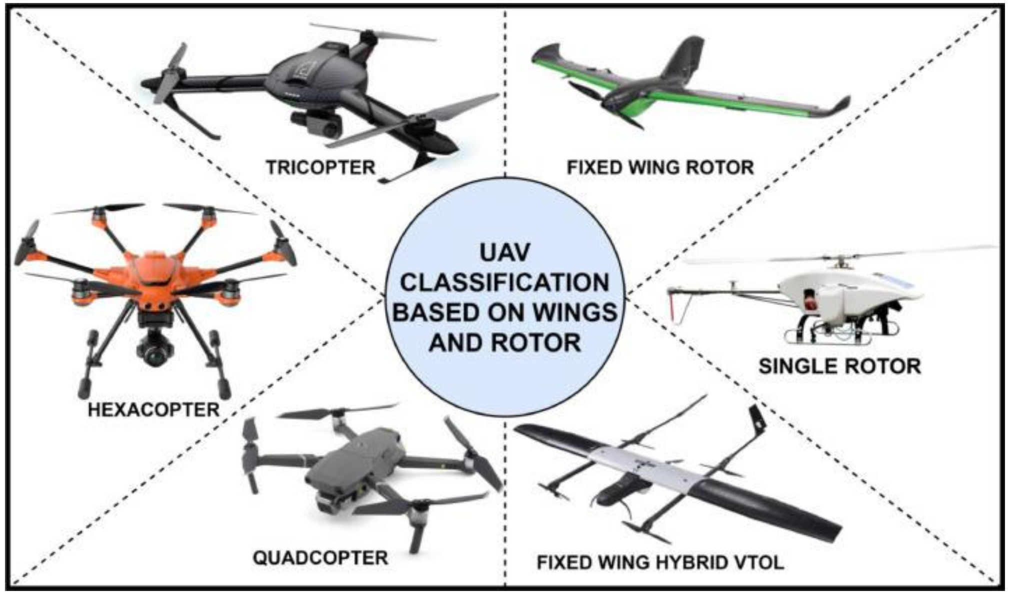
System of systems

B737MAX accidents in 2019

Consensus n. Collective opinion or concord; general agreement or accord. [Latin, from *consentire,* to agree]

UAS vs Drone

- **UAS(Unmanned Aircraft Systems) -International**
  - UAS are air vehicles and associated equipment that do not carry a human operator, but instead are remotely piloted or fly autonomously. UAS commonly are referred to as Unmanned Aerial Systems (UAS), Unmanned Aerial Vehicles (UAV), Remotely Piloted Aircraft Systems (RPAS) and drones.
  - A UAS generally consists of
    1) an aircraft with no pilot on board,
    2) a remote pilot station,
    3) a command and control link
    4) a payload specific to the intended application/operation

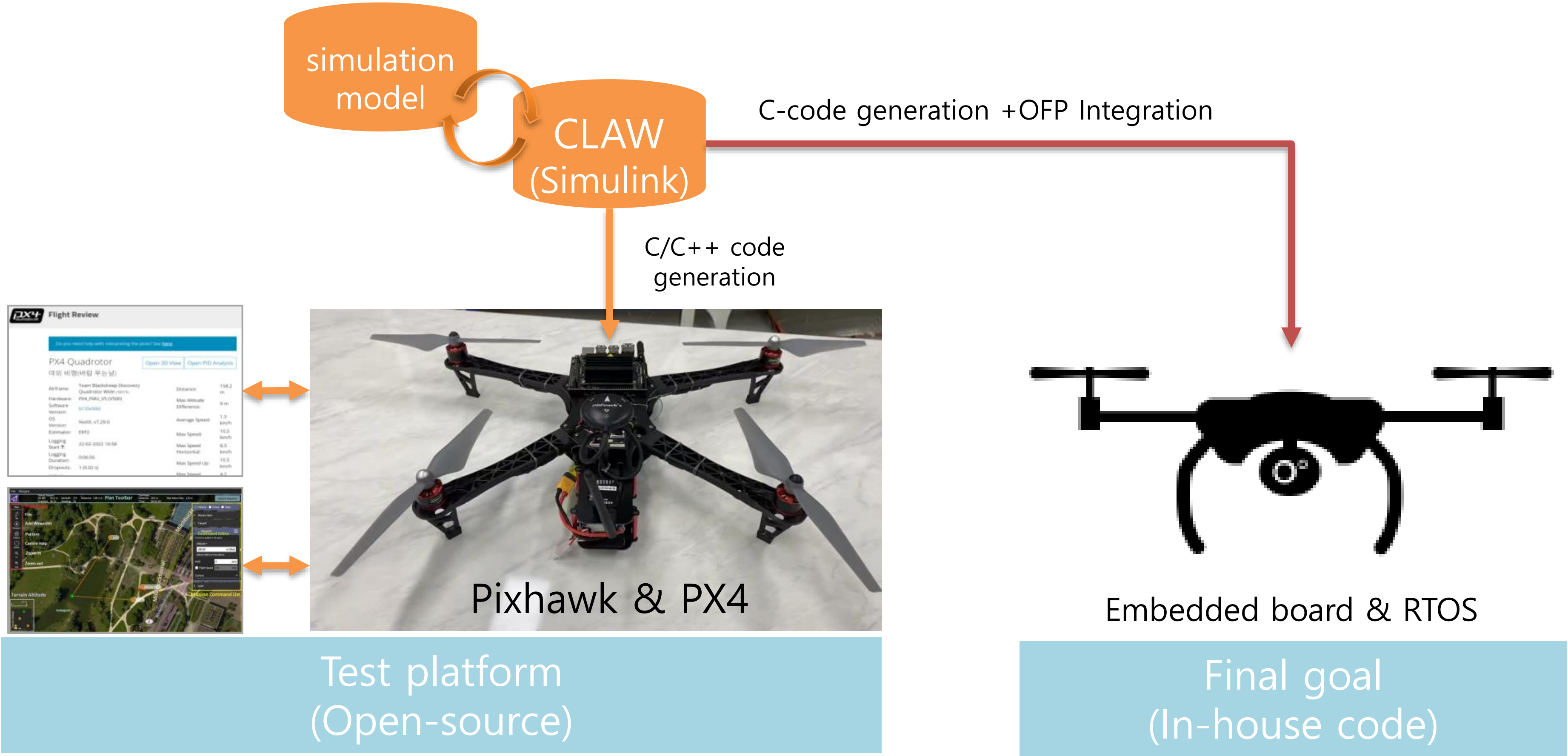


- **Drone**
  - An aircraft without a pilot that is controlled by someone on the ground, used especially as a hobby
  - A low, continuous noise
  - In this presentation, drone refer only to multi-copter type UAS

- **AAM(Advanced Air Mobility)**
  - AAM is an air transport system concept that integrates new, transformational aircraft designs and flight technologies into existing and modified airspace operations. - NASA
  - Urban Air Mobility(UAM) envisions a safe and efficient aviation transportation system that will use highly automated aircraft that will operate and transport passengers or cargo at lower altitudes within urban and suburban areas. - FAA

Research goal

- Development of a flyable flight control software for drone(multi-copter) using Model Based Development tools



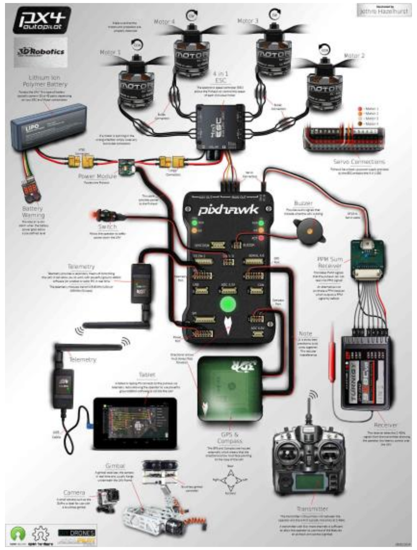
CLAW : Control Law
OFP : Operational Flight Program
RTOS : Real-time Operating System

Problem of UAV control engineer

- **Different development environment between flight control engineer & software engineer**
  - Aerospace engineers are NOT familiar with Linux and code developing environment, but interested in the control algorithm & logics
- **Embedded source code(C/C++) is dependent with hardwares(DSP, MPC,FPGA,...)**
- **The development period is getting shorter; Customers don't wait until next week(Share it now through YouTube)**
  - Many drone open-source communities and resources that have emerged recently
    - Pixhawk, PX4, Ardupilot, QGCS, UGCS, MAVlink, FlightReview, Gazebo, Airsim(Unreal Engine) and ROS

Aerospace Engineer

Pixhawk (Open-source drone platform)



Software Engineer

Purpose & approach

Purpose

- Development of in-house **Control Law(CLAW)** for multi-copter independent of hardware & operating system
  - CLAW with good performance & handling quality compatible with DJI drone
  - Efficient software development & testing environments using open-source & commercial tools
  - Software verification & code quality level(C/C++ code)
  - Software configuration management and expandability
  - Convenient on-board system testing with UAV/Drone

"CLAW to compete with DJI Mavic 2"

Project Approach

- Model-Based Development(MBD) using auto C/C++ code generation
- CLAW model verification using Model advisor, dynamic model
- CLAW Project management with Simulink Project
- Visualization with commercial tools(X-Plane, Gazebo,...)
- Pixhawk open-hardware testing with PX4 support package

[Model-Based Development applications for UAV]

Ref.1 J.H.Moon, S.H.Oh, J.K.Kim, and I.J .Chung, Development of the synchropter control law using the open-source platform and the model-based-design software, Proceeding of the 2020 KSAS Spring Conference, 2020, pp.535-536.
Ref.2 Korean Air Speeds UAV Flight Control Software Development ..., https://kr.mathworks.com/company/user_stories/korean-air-speeds-uav-flight-control-software-development-and-verification-with-model-based-design.html

Drone open-source environment

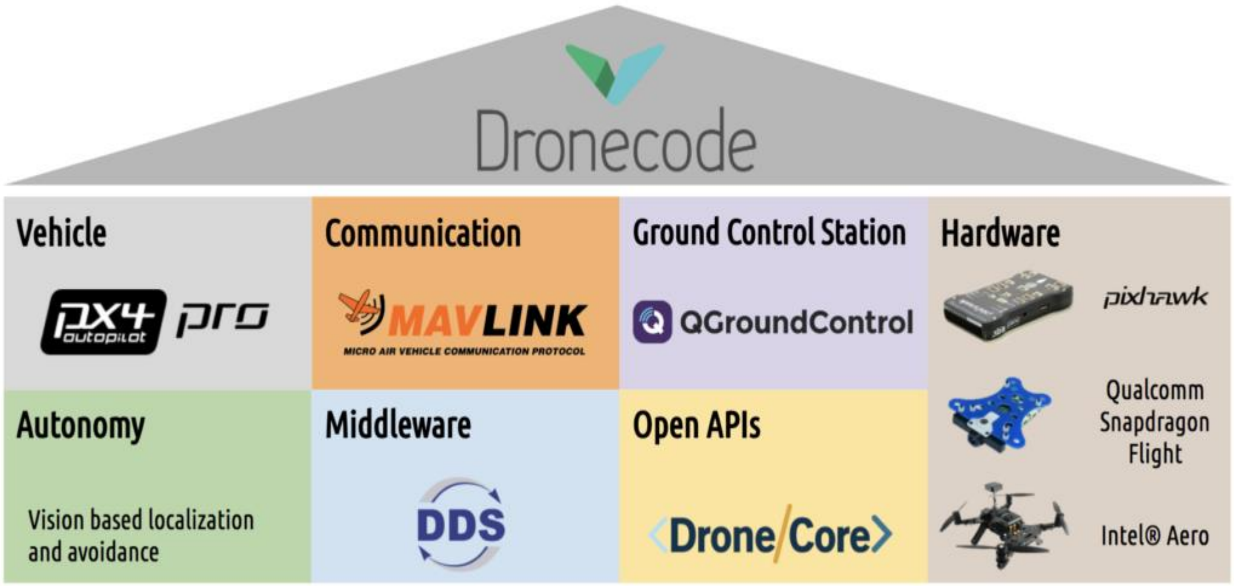
- **Hardware : Pixhawk open-architecture flight control board**
- **Software : Open-source autopilot system oriented toward inexpensive autonomous aircraft**
  - Ardupilot : Arduino based open-source autopilot (It supports various types of aircraft)
  - PX4 : Software oriented open-source autopilot (It supports various middleware such as ROS, DDS, etc)
    - Project started in 2009, Initial release 2012
    - Further developed and used at Computer Vision & Geometry Lab of ETH
    - Supported by Autonomous Systems Lab and Automatic Control Lab
    - Language : C++
    - Operating System : NuttX, ROS
    - License : BSD

PX4
- Optimized for multicopter
- Specialized for integration with vision based processing
- Support ROS, DDS
- GCS : QGCS,
- BSD License

Ardupilot
- Support various types of aircraft fixed-wing, helicopter, multicopter(more options)
- GCS : Mission planner
- GNU License

MAVLINK

Open-architecture hardware



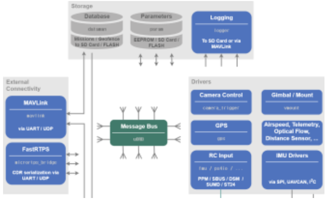
Development environment of drone controller using PX4

Autonomous System Research Group

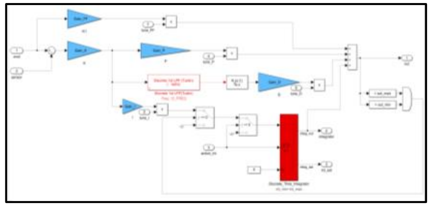
Companion board(nVidia Jetson)

Advantage
- Expandability with other software
- Easy to integrate with vision processing

Disadvantage
- Weight & power consumption
- Required Linux development environment
- Engineer should integrate code with Ubuntu
- Limited to access low-level controller



PX4



Customized controller

Flight Control board

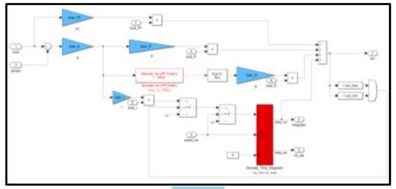
UART
Offboard mode

Companion board

C/C++ Code level integration with PX4

Advantage
- Fully support customized interface
- Easy to use existing PX4 module
- Easy to update PX4 firmware

Disadvantage
- Require more steps to make firmware
- Engineer should know PX4 code related with task
- PX4 code change according to the Simulink model



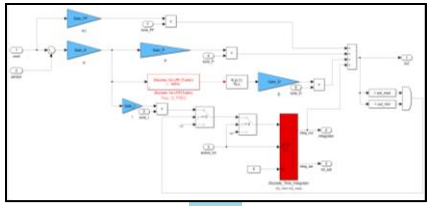
C code generation

Integration with PX4 source code

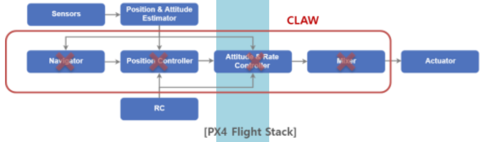
PX4 Support package

Advantage
- Easy to make firmware (One click)
- Does not required additional C/C++ code

Disadvantage
- Limited access to variables of PX4
- Limited firmware version(v1.10.2 only)
- Limited hardware supports



C/C++ code generation with task

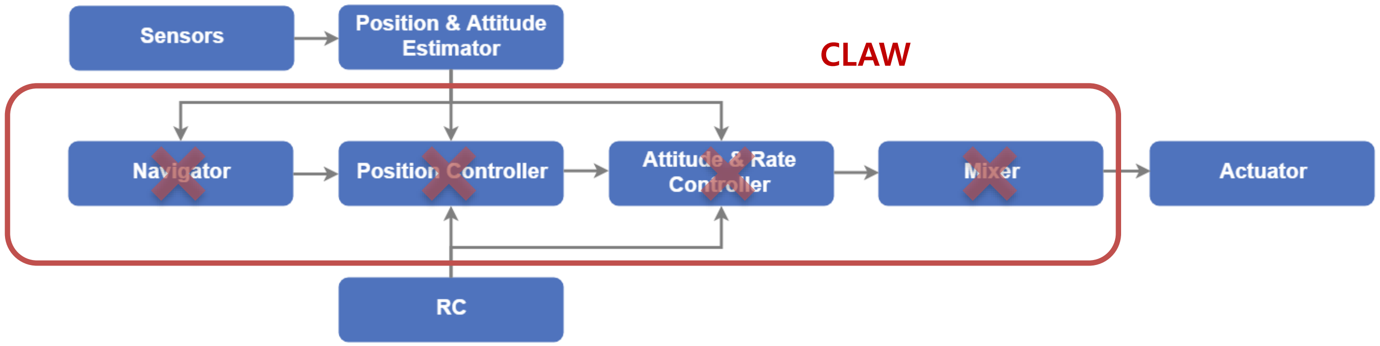


Compile & targeting through
UAV Toolbox PX4 Support package

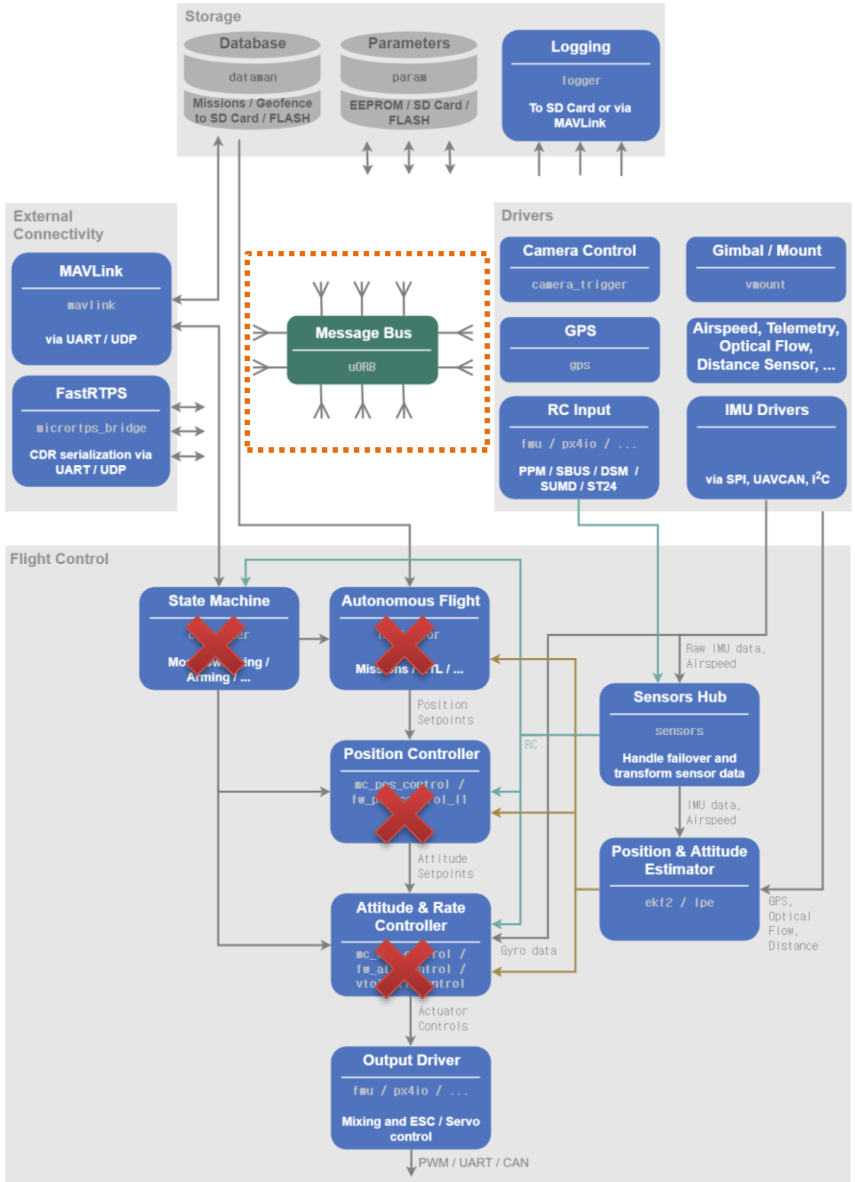
PX4 software architecture

Autonomous System Research Group

- PX4 Software architecture
  - All functionality is divided into exchangeable and reusable components
  - Communication is done by asynchronous message passing (uORB)
  - The system can deal with varying workload (NuttX)
- CLAW modules (→ replace with Simulink model)
  - Commander : Manage flight/control mode
  - Navigator : Waypoint guidance
  - Position controller : Maintain and follow position command
  - Attitude/rate controller : Stabilize and control attitude
  - Mixer : Command motor output(PWM)



[PX4 Flight Stack]



[High-level Software architecture]

PX4 uORB message

Autonomous System Research Group

- **Interface with PX4 through uORB messages**
  - uORB is asynchronous publish() / subscribe() messaging API used for inter-thread/inter-process communication.
  - All message contains timestamp(μsec) because of asynchronous communication
  - Each module publishes and subscribes topic related to special purpose
    - PX4 topic : 117ea (Firmware v1.10.2)

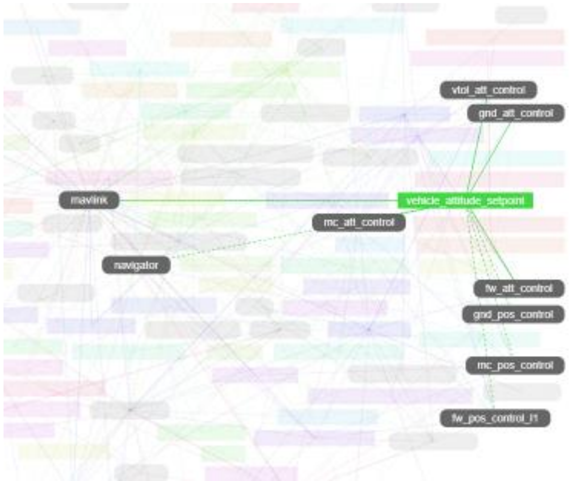
Topic : vehicle_attitude_setpoint

Contents	Description
timestamp	Time since system start(μsec)
roll_body	X-body angle in NED frame
pitch_body	Y-body angle in NED frame
yaw_body	Z_body angle in NED frame
yaw_sp_move_rate	Yaw rate command
q_d[4]	Desired quaternion
q_d_valid	Set to true if quaternion is valid
thrust_body[3]	Normalized thrust command
roll_reset_integral	Reset roll integral
pitch_reset_integral	Reset pitch integral
yaw_reset_integral	Reset yaw integral
....	Fixed wing UAV variables

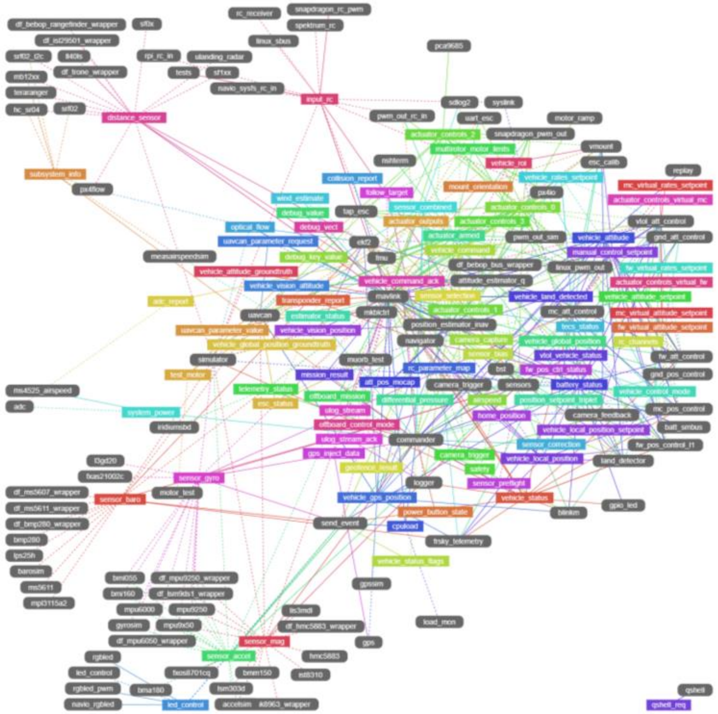
Publisher → Subscriber

Publisher:
- mc_pos_control
- navigator

Subscriber:
- mc_att_control
- mavlink

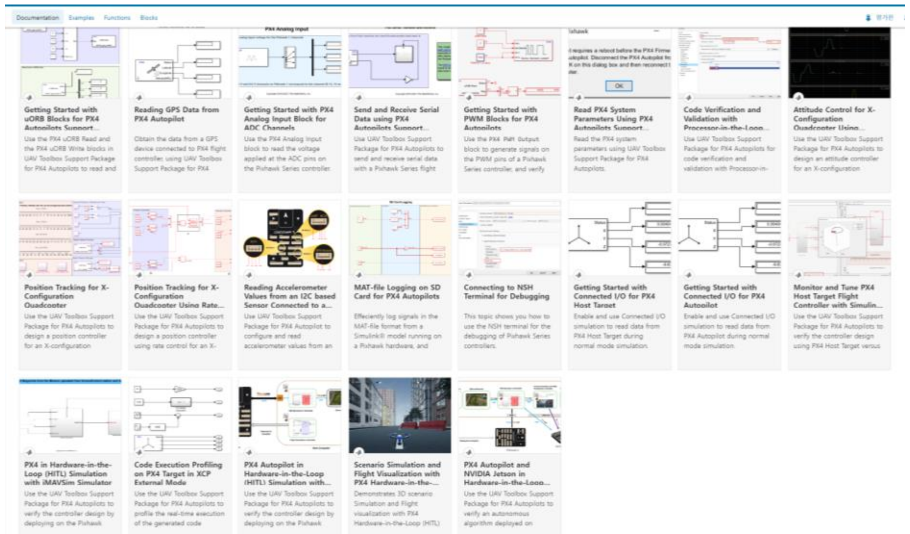


[uORB Graph]

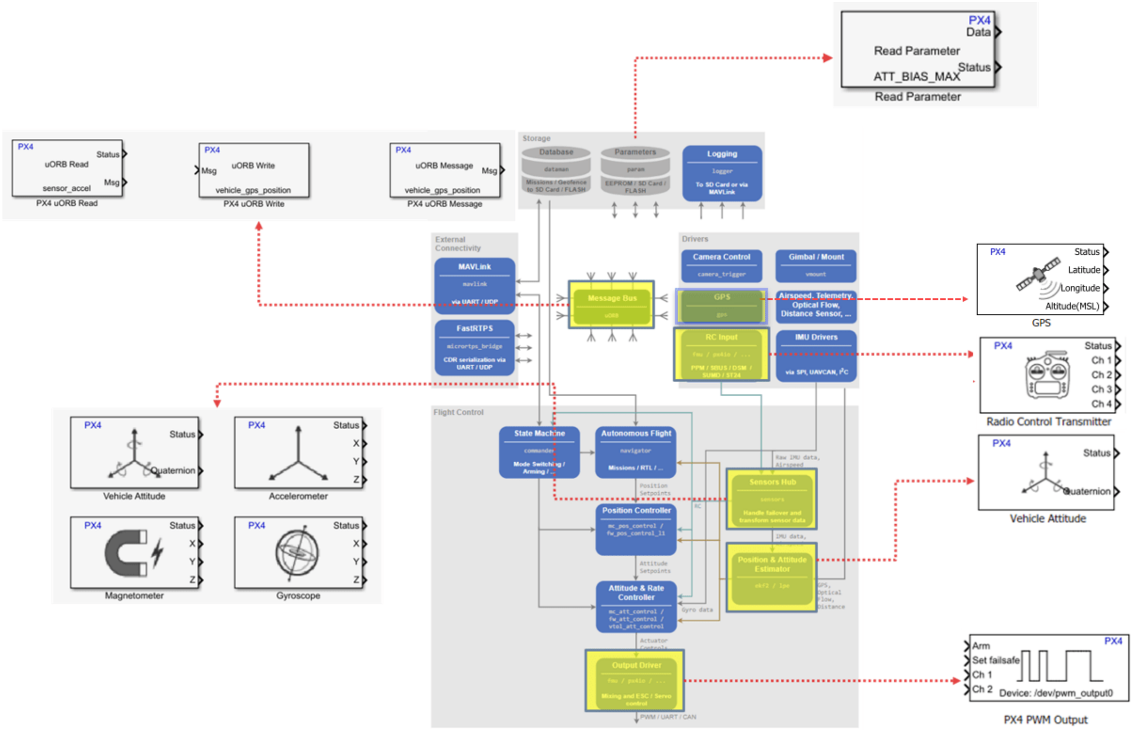


MATLAB PX4 Support package

- **UAV Toolbox & PX4 Support package**
  - ➢ The PX4 support package provides blocks such as reading parameters and reading/writing uORB messages
    - Customized blocks : RC input, vehicle attitude, accelerometer, etc
    - PWM output driver blocks
    - Serial communication blocks



[UAV Toolbox PX4 Support package]



[PX4 support blocks]

MATLAB PX4 Support package

- **PX4 Task**
  - Px4_Simulink_app_task : customized CLAW task

Custom CLAW

Pre Flight:

```
PID COMMAND                    CPU(ms) CPU(%)  USED/STACK PRIO(BASE) STATE  FD
  0 Idle Task                    28771 38.100   216/  512   0 (  0)  READY   3
  1 hpwork                           0  0.000   344/ 1260 249 (249)  w:sig   3
  2 lpwork                          34  0.100   688/ 1516  50 ( 50)  w:sig   8
  3 init                           902  0.000  2072/ 2604 100 (100)  w:sem   3
  4 wq:manager                       0  0.000   368/ 1252 243 (243)  w:sem   4
 80 wq:att_pos_ctrl               3517  6.100  4768/ 6596 244 (244)  w:sem   4
 14 wq:lp_default                   19  0.100   812/ 1700 250 (205)  w:sem   4
 18 wq:I2C1                        219  0.400   848/ 1244 248 (248)  w:sem   4
 22 wq:hp_default                  601  1.100  1076/ 1596 243 (243)  w:sem   4
 24 px4io                          556  0.900   916/ 1484 240 (240)  w:sem   4
 27 gps                            124  0.200   960/ 1644 208 (208)  w:sem   4
 30 wq:SPI1                      12575 21.600  1016/ 1396 254 (254)  w:sem   4
 47 wq:I2C3                        198  0.400   872/ 1244 246 (246)  w:sem   4
 49 wq:SPI4                        189  0.300   600/ 1396 251 (251)  w:sem   4
 77 sensors                       1534  2.600  1308/ 1964 237 (237)  w:sem  10
 83 wq:rate_ctrl                  1004  1.800   892/ 1596 255 (255)  w:sem   4
233 log_writer_file                184 13.300   824/ 1164  60 ( 60)  w:sem   4
191 mavlink_if0                    762  1.200  1648/ 2484 100 (100)  w:sig   4
197 mavlink_rcv_if0                179  0.300  2520/ 3916 175 (175)  w:sem   4
221 px4_simulink_app_task            1  0.000   816/ 2012 205 (205)  w:sem  16
230 mavlink_if1                      4  0.000  1384/ 2532 100 (100)  w:sig   3
232 logger                         289  1.600  3088/ 3668 233 (233)  RUN     4
249 <pthread>                     6032  8.300   752/ 8700 250 (250)  w:sem  16
250 <pthread>                       60  0.100   368/ 8700 249 (249)  w:sem  16

Processes: 24 total, 2 running, 22 sleeping, max FDs: 20
CPU usage: 60.40% tasks, 1.50% sched, 38.10% idle
DMA Memory: 5120 total, 1024 used 1024 peak
Uptime: 59.088s total, 28.771s idle
```

Original PX4

Pre Flight:

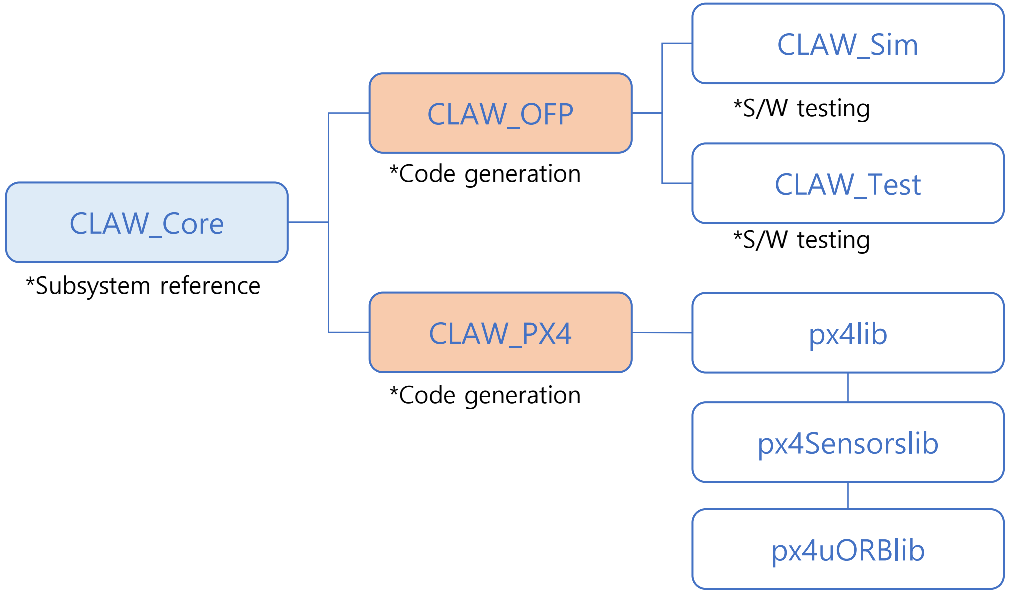
```
PID COMMAND                    CPU(ms) CPU(%)  USED/STACK PRIO(BASE) STATE  FD
  0 Idle Task                  1900091 44.000   244/  512   0 (  0)  READY   3
  1 hpwork                           0  0.000   344/ 1260 249 (249)  w:sig   3
  2 lpwork                        2272  0.000  1016/ 1516  50 ( 50)  w:sig   8
  3 init                           871  0.000  1992/ 2604 100 (100)  w:sem   3
  4 wq:manager                       0  0.000   384/ 1252 243 (243)  w:sem   4
187 wq:att_pos_ctrl             235324  6.500  4768/ 6596 244 (244)  w:sem   4
 17 dataman                         57  0.000   752/ 1180  90 ( 90)  w:sem   4
 20 wq:lp_default                 2622  0.000   812/ 1700 205 (205)  READY   4
 24 wq:I2C1                      17824  0.400   936/ 1244 248 (248)  w:sem   4
 28 wq:hp_default                35945  1.000  1120/ 1596 243 (243)  w:sem   4
142 wq:SPI1                     803364 22.200   960/ 1396 254 (254)  w:sem   4
157 wq:I2C3                      12533  0.400   824/ 1244 246 (246)  w:sem   4
159 wq:SPI4                      12152  0.300   600/ 1396 251 (251)  w:sem   4
186 sensors                      96359  2.400  1308/ 1964 237 (237)  w:sem  10
197 commander                    27000  0.800  2160/ 3212 140 (140)  w:sig   6
198 wq:rate_ctrl                216989  6.200  1120/ 1596 255 (255)  w:sem   4
199 commander_low_prio              57  0.000   608/ 2996  50 ( 50)  w:sem   6
277 gps                           5730  0.200  1088/ 1620 208 (208)  w:sem   4
335 mavlink_if1                  38861  1.100  1688/ 2484 100 (100)  READY   5
336 mavlink_rcv_if1              10661  0.300  2792/ 3916 175 (175)  w:sem   5
371 px4io                       112950  3.200  1048/ 1484 240 (240)  w:sem   4
651 navigator                     2354  0.000   896/ 1764 105 (105)  READY   4
695 logger                       12833  1.600  3080/ 3644 233 (233)  RUN     4
702 log_writer_file                295  7.200   824/ 1164  60 ( 60)  w:sem   4

Processes: 24 total, 5 running, 19 sleeping, max FDs: 20
CPU usage: 53.80% tasks, 2.20% sched, 44.00% idle
DMA Memory: 5120 total, 2048 used 2048 peak
Uptime: 3618.400s total, 1900.091s idle
```

CLAW Software development for multicopter

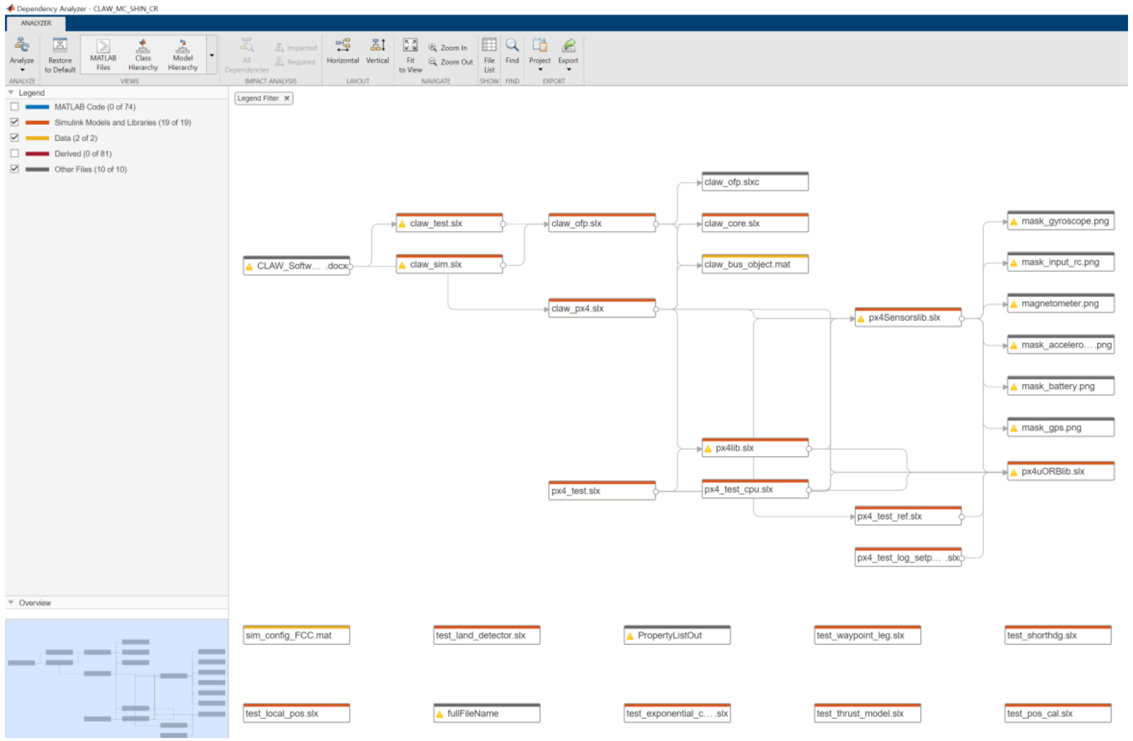
CLAW S/W Design

CLAW development project

- Hardware independent control law development environments
  - CLAW Core : control law mode and controller (subsystem reference)
  - CLAW OFP : code generation settings and input/output definition to integrated with custom OFP
  - CLAW PX4 : input/output definitions and logics to interface with PX4
- Software test model
  - CLAW_Sim : flight dynamics model for controller verification
  - CLAW_Test : unit test model of CLAW



[Simulink model dependency tree]

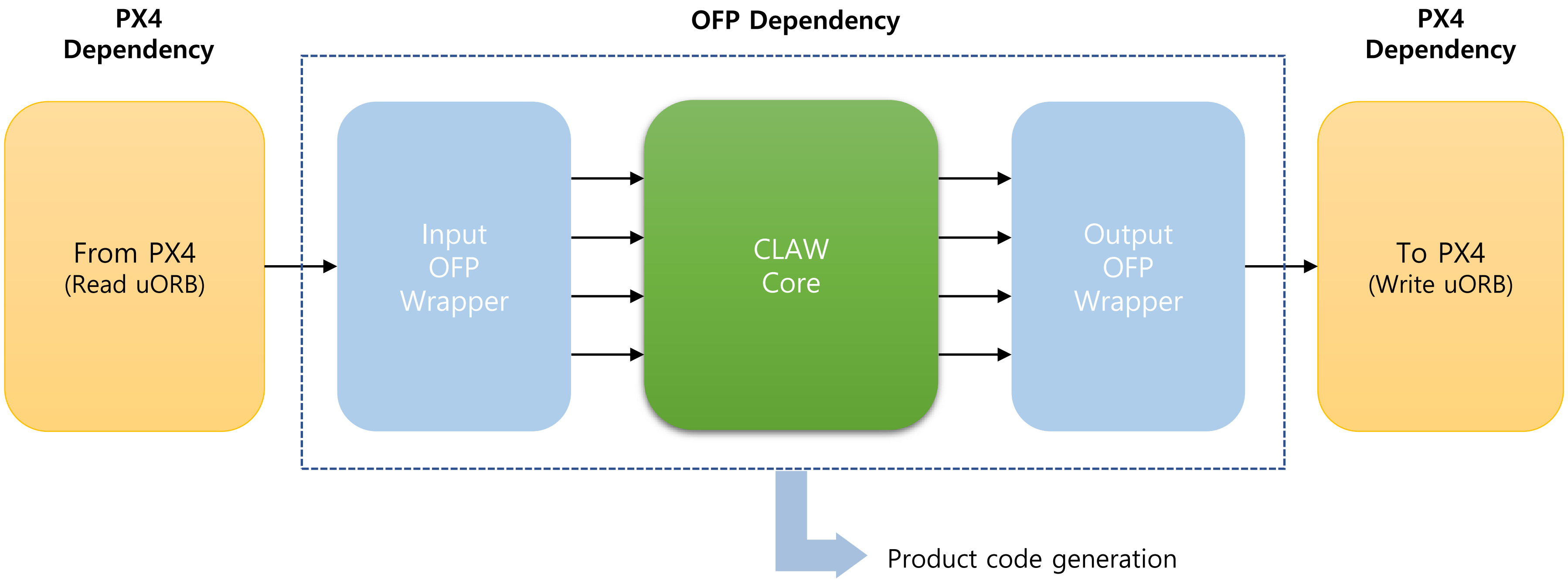


[Project dependency analyzer]

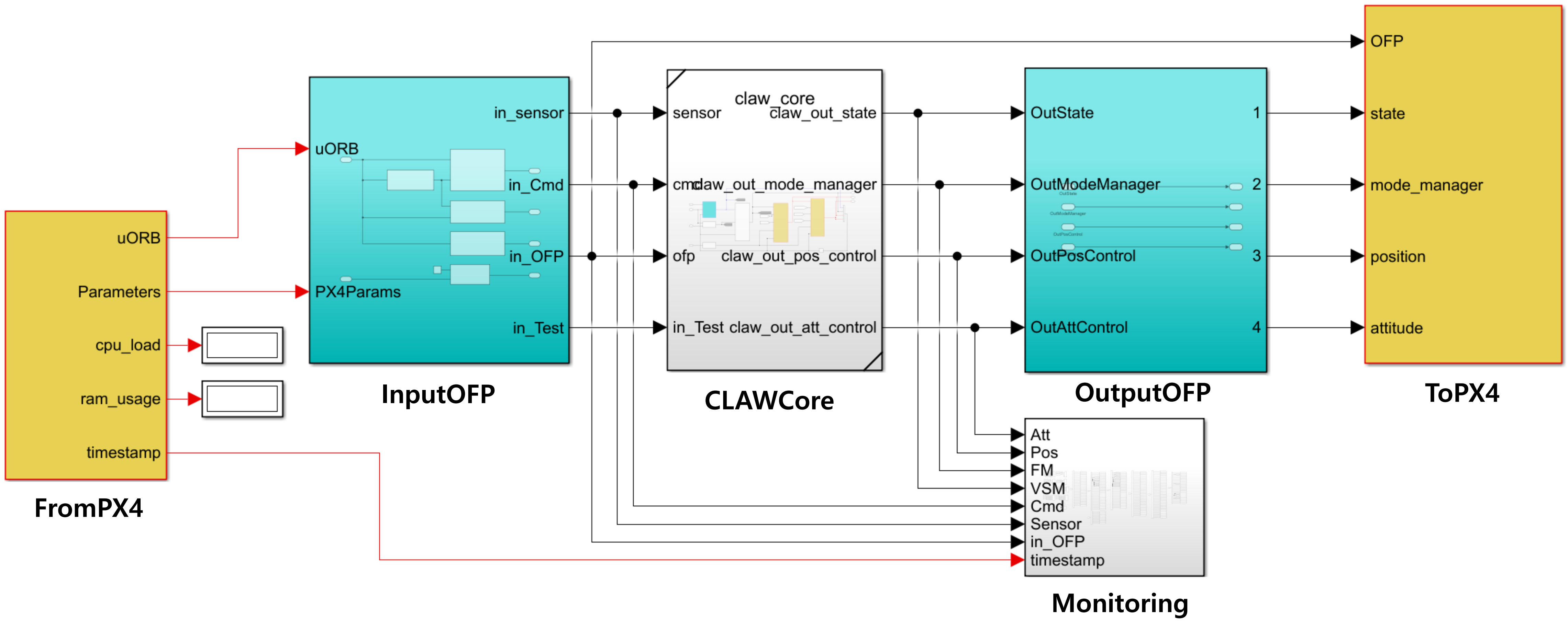
CLAW S/W Design

CLAW_PX4 Simulink model architecture

- From PX4 : Receive navigation information and sensor such as attitude, angular rate, accelerations from uORB message
- CLAW Core : Sensor filtering, flight decision, flight mode management, position/velocity control, attitude control
- Input/Output Wrapper : Interface between OFP and PX4
- To PX4 : Output motor command and CLAW status



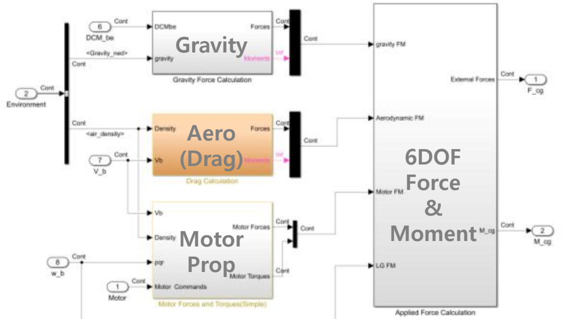
- Top-level Simulink model of the CLAW for PX4 integration



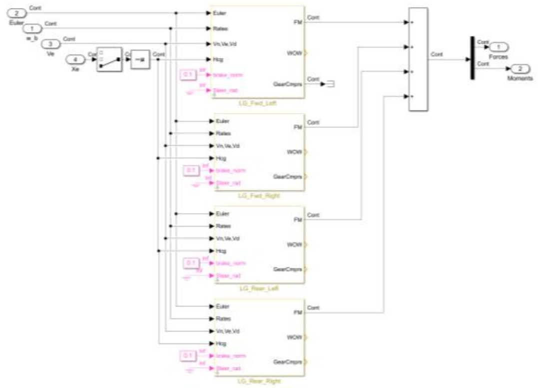
CLAW S/W Design

- **Simulation & verification environment in MATLAB/Simulink**
  - The motor/prop model considers effects of motor/arm canted angles($\emptyset, \theta, \psi$) and installation position(x, y, z)
  - The landing gear model includes four spring-damper system
  - The aerodynamic model contains only the drag term (Trim attitude effect in forward/lateral flight)

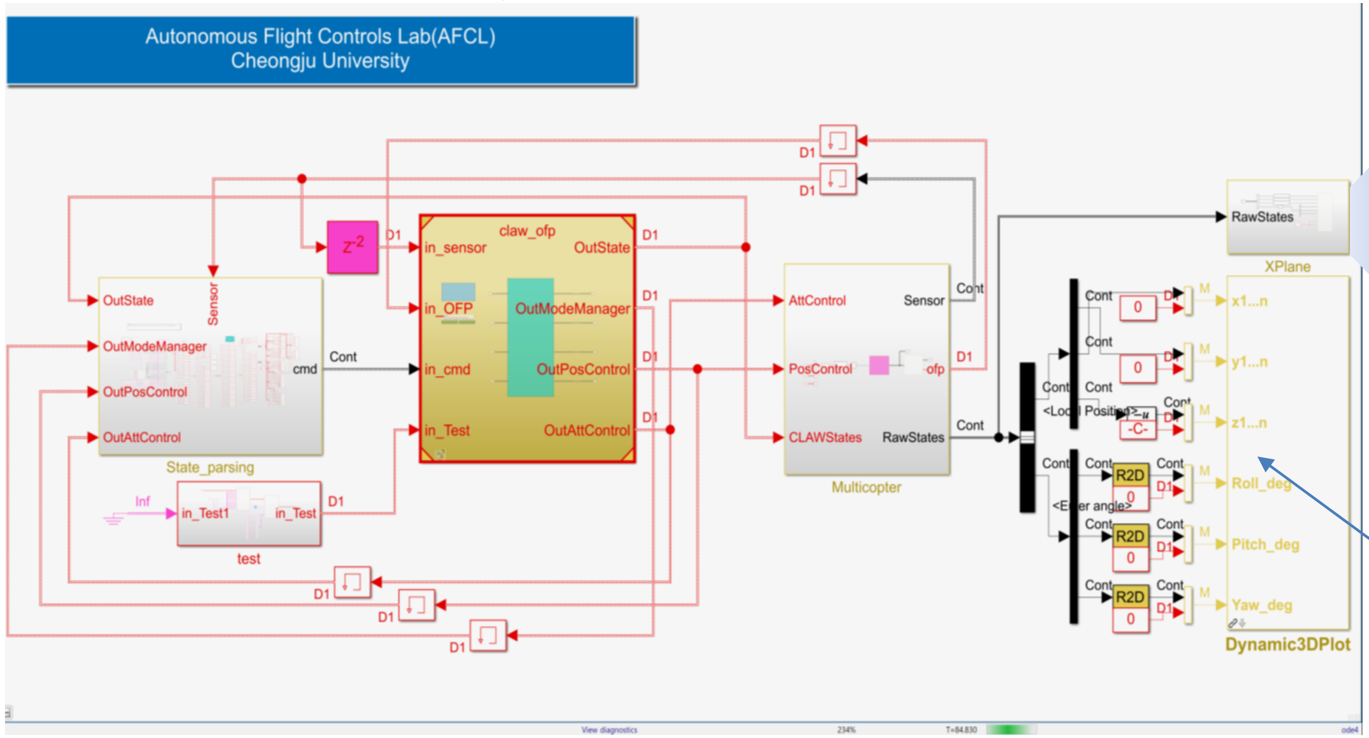
Force & Moments model



Landing gear model

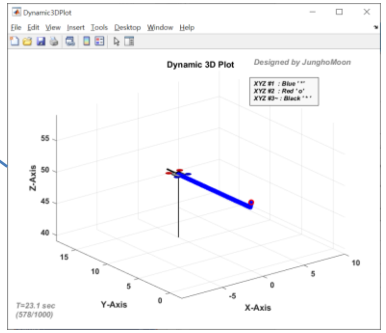


Full flight simulation model (CLAW_sim)



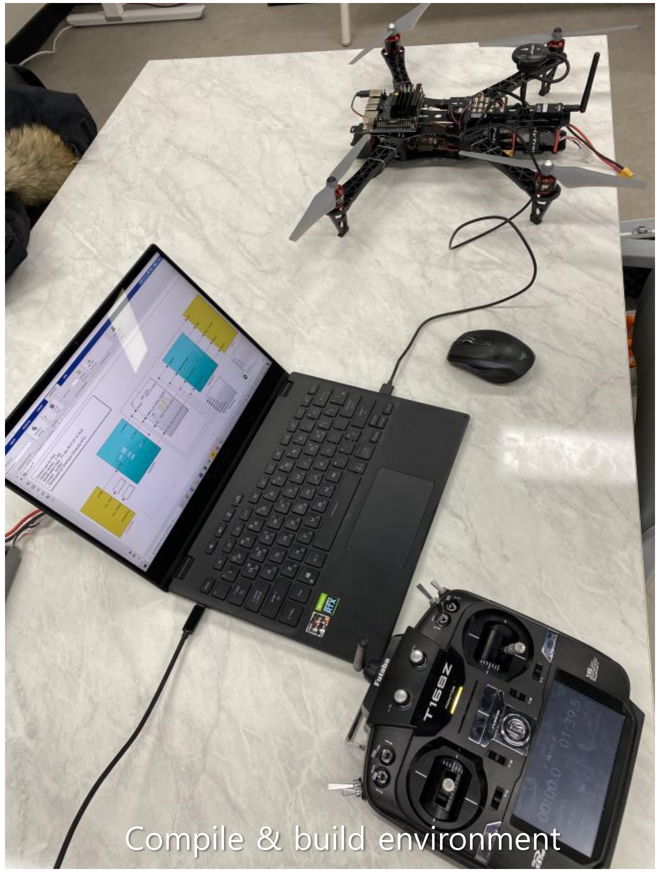
X-Plane(UDP)

Realtime 3D plot



Flight test

- **CLAW was verified through indoor & outdoor flight test**
  - Purpose : CLAW function and performance verification through flight test
  - Platform : TBS-500 Quadcopter, Pixhawk4, PX4 v1.10.2 (customized CLAW task)
  - GCS : Windows10, QGCS, Futaba T16SZ R/C Controller, 433MHz UHF modem
  - Location : Cheongju University, Indoor flight test lab

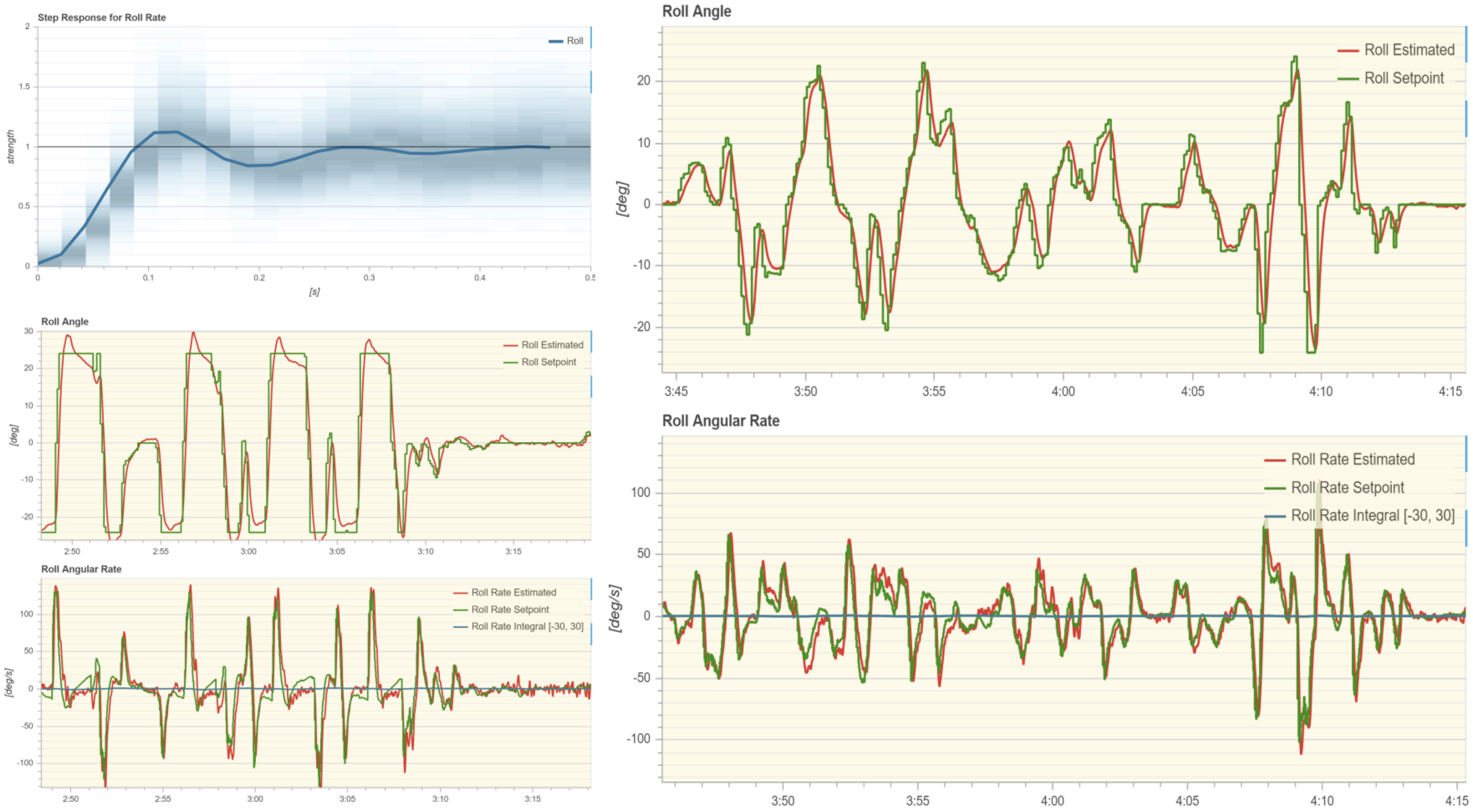


Compile & build environment

Flight test

- **Flight data monitoring & analysis using PX4 tools**
  - QGCS : Open-source ground control software for drone
  - Flight Review : Open-source flight data analyzer that is easy to use and powerful in performance analysis

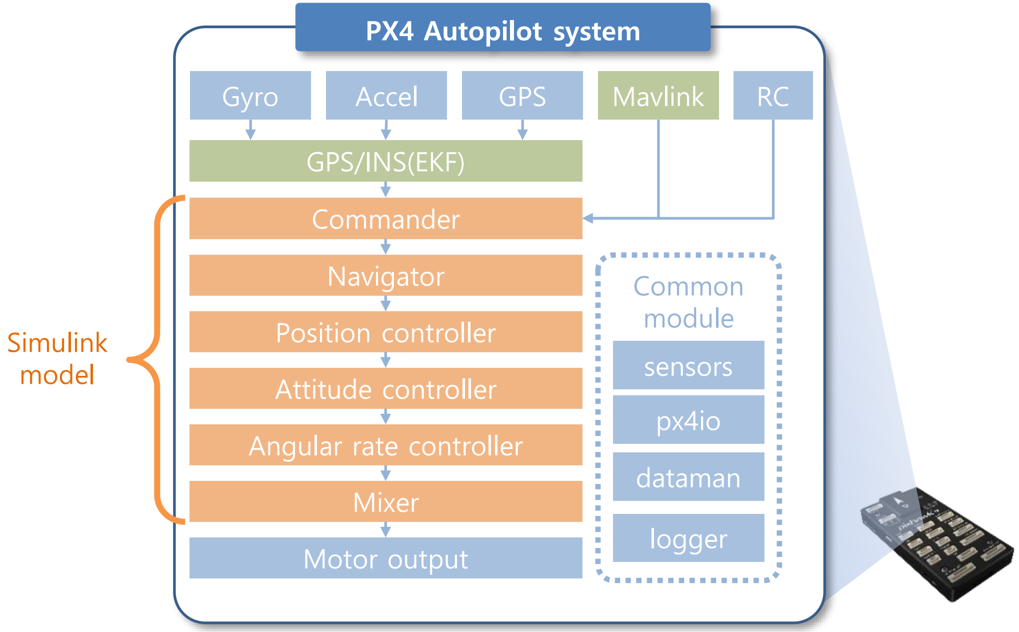
(Flight Review®)PID Analysis



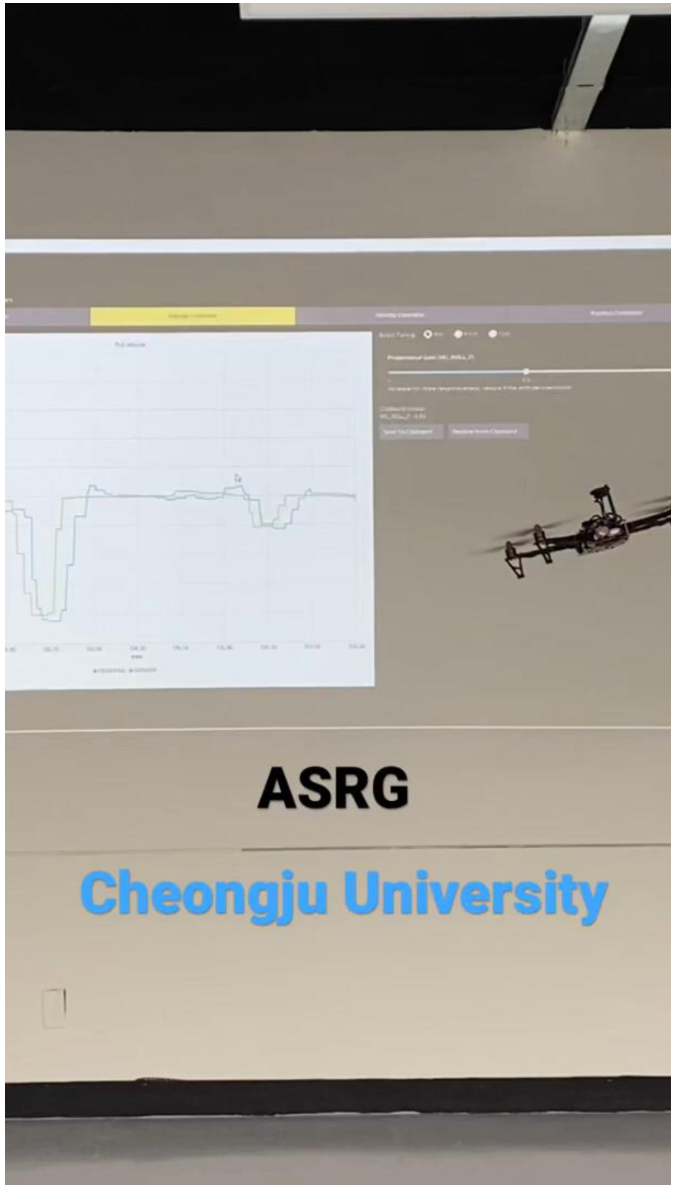
Conclusion

- **Result**
  - We analyzed PX4 module architecture and uORB for CLAW&PX4 integration
  - The common CLAW software framework was designed for custom OFP and PX4
  - CLAW was verified through flight tests with open architecture hardware & software(Pixhawk+PX4)

- **Future work**
  - Simulink/MAVLINK interface module development
  - Aggressive maneuvering controller design & testing (H-infinity, MPC, L1 adaptive control)
  - GPS/INS(EKF) filter design and implementation using Simulink



Youtube link



Thank you