MATLAB과 PX4를 활용한 드론 비행제어 소프트웨어 개발
Development of drone flight control software using MATLAB and PX4
문정호 교수, 청주대학교 무인항공기학과
Introduction
- Autonomous System Research Group (ASRG)
- Research goal & problem
- PX4 open-source flight control software
- uORB message
- UAV Toolbox & PX4 Support package

CLAW Software development
- Simulink project & model architecture
- Attitude/Position controller
- CLAW additional logics
- Simulation model
- Flight test result

Conclusion & Future work
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

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)

Speciality of research team

1st Controls for autonomous aircraft
2nd Machine learning for aerospace
3rd Convergence of robotics & aerospace engineering
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

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

**Research goal**

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

- Simulation model: CLAW (Simulink)  
- C-code generation + OFP Integration  
- C/C++ code generation

**Test platform (Open-source):** Pixhawk & PX4

**Final goal (In-house code):** Embedded board & RTOS

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

Software Engineer

Pixhawk (Open-source drone platform)
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

**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

"CLAW to compete with DJI Mavic 2"

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

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

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

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
PX4 software architecture

- **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)
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

<table>
<thead>
<tr>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>Time since system start(μsec)</td>
</tr>
<tr>
<td>roll_body</td>
<td>X-body angle in NED frame</td>
</tr>
<tr>
<td>pitch_body</td>
<td>Y-body angle in NED frame</td>
</tr>
<tr>
<td>yaw_body</td>
<td>Z_body angle in NED frame</td>
</tr>
<tr>
<td>yaw_sp_move_rate</td>
<td>Yaw rate command</td>
</tr>
<tr>
<td>( \mathbf{q}_{d}[4] )</td>
<td>Desired quaternion</td>
</tr>
<tr>
<td>( \mathbf{q}_{d} \text{valid} )</td>
<td>Set to true if quaternion is valid</td>
</tr>
<tr>
<td>thrust_body[3]</td>
<td>Normalized thrust command</td>
</tr>
<tr>
<td>roll_reset_integral</td>
<td>Reset roll integral</td>
</tr>
<tr>
<td>pitch_reset_integral</td>
<td>Reset pitch integral</td>
</tr>
<tr>
<td>yaw_reset_integral</td>
<td>Reset yaw integral</td>
</tr>
<tr>
<td>...</td>
<td>Fixed wing UAV variables</td>
</tr>
</tbody>
</table>

[uORB Graph]
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
## PX4 Task

- **Px4_Simulink_app_task**: customized CLAW task

### Custom CLAW

**Pre Flight:**

<table>
<thead>
<tr>
<th>PID COMMAND</th>
<th>CPU(ms)</th>
<th>CPU(%)</th>
<th>USED/STACK</th>
<th>PRIO/Base</th>
<th>STATE FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Idle Task</td>
<td>28771</td>
<td>38.100</td>
<td>216/512</td>
<td>0 (0)</td>
<td>READY 3</td>
</tr>
<tr>
<td>1 sppwork</td>
<td>0.000</td>
<td>0.000</td>
<td>344/1260</td>
<td>249 (249)</td>
<td>w:sig 3</td>
</tr>
<tr>
<td>2 lipwork</td>
<td>34.100</td>
<td>0.000</td>
<td>688/1516</td>
<td>58 (50)</td>
<td>w:sig 8</td>
</tr>
<tr>
<td>3 iinit</td>
<td>902.000</td>
<td>0.000</td>
<td>2672/2604</td>
<td>100 (100)</td>
<td>w:sem 3</td>
</tr>
<tr>
<td>4 wq:manager</td>
<td>0.000</td>
<td>0.000</td>
<td>568/1252</td>
<td>243 (243)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>80 wq:att_pos_ctl</td>
<td>3517.600</td>
<td>4.100</td>
<td>4768/6596</td>
<td>244 (244)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>14 wq:lp_default</td>
<td>19.100</td>
<td>0.200</td>
<td>812/1700</td>
<td>250 (250)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>18 wq:i2c1</td>
<td>219.400</td>
<td>2.600</td>
<td>848/1244</td>
<td>248 (248)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>22 wq:hp_default</td>
<td>601.100</td>
<td>7.200</td>
<td>1076/1596</td>
<td>243 (243)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>24 px4io</td>
<td>556.900</td>
<td>6.500</td>
<td>916/1484</td>
<td>240 (240)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>27 gps</td>
<td>124.200</td>
<td>1.300</td>
<td>960/1644</td>
<td>208 (208)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>38 wq:spi1</td>
<td>1257.500</td>
<td>15.300</td>
<td>2168/1396</td>
<td>254 (254)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>47 wq:i2c3</td>
<td>198.400</td>
<td>2.300</td>
<td>872/1244</td>
<td>246 (246)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>49 wq:spi4</td>
<td>189.300</td>
<td>2.200</td>
<td>608/1396</td>
<td>251 (251)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>77 sensors</td>
<td>1534.200</td>
<td>18.700</td>
<td>1388/1964</td>
<td>237 (237)</td>
<td>w:sem 10</td>
</tr>
<tr>
<td>83 wq:rate_ctl</td>
<td>1004.100</td>
<td>12.000</td>
<td>892/1596</td>
<td>255 (255)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>233 log_writer_file</td>
<td>184.1300</td>
<td>2.000</td>
<td>824/1164</td>
<td>60 (60)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>191 navlink_if0</td>
<td>762.100</td>
<td>8.600</td>
<td>1648/2484</td>
<td>100 (100)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>197 navlink_rcv_if0</td>
<td>379.3000</td>
<td>4.200</td>
<td>2528/3916</td>
<td>175 (175)</td>
<td>w:sem 4</td>
</tr>
</tbody>
</table>

**221 px4_simulink_app_task**

- CPU(ms): 1.000
- CPU(%) : 0.000
- STATE FD : 216/2021 205 (205)
- CPU: 0.000
- USED/STACK: 2532 250 (250)
- CPU: 16
- USED/STACK: 20

**230 navlink_if1**

- CPU(ms): 4.000
- CPU(%) : 0.000
- STATE FD : 1384/2532 250 (250)
- CPU: 4
- USED/STACK: 0
- CPU: 0
- USED/STACK: 0

**232 logger**

- CPU(ms): 289.100
- CPU(%) : 0.000
- STATE FD : 3688/3668 233 (233)
- CPU: 0
- USED/STACK: 0
- CPU: 0
- USED/STACK: 0

**249 <px_thread>**

- CPU(ms): 632.800
- CPU(%) : 0.000
- STATE FD : 752/8700 250 (250)
- CPU: 0
- USED/STACK: 0
- CPU: 0
- USED/STACK: 0

**258 <px_thread>**

- CPU(ms): 60.100
- CPU(%) : 0.000
- STATE FD : 368/8700 249 (249)
- CPU: 0
- USED/STACK: 0
- CPU: 0
- USED/STACK: 0

**Processes:** 24 total, 2 running, 22 sleeping, max FDs: 20

**CPU usage:** 68.48% tasks, 1.50% sched, 38.10% idle

**Memory Usage:** 5120 total, 1024 used 1024 peak

**Uptime:** 59.088s total, 28.771s idle

---

## MATLAB PX4 Support package

- **PX4 Task**
- **Original PX4**

**Pre Flight:**

<table>
<thead>
<tr>
<th>PID COMMAND</th>
<th>CPU(ms)</th>
<th>CPU(%)</th>
<th>USED/STACK</th>
<th>PRIO/Base</th>
<th>STATE FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Idle Task</td>
<td>1900091</td>
<td>44.000</td>
<td>244/512</td>
<td>0 (0)</td>
<td>READY 3</td>
</tr>
<tr>
<td>1 sppwork</td>
<td>0.000</td>
<td>0.000</td>
<td>346/1260</td>
<td>249 (249)</td>
<td>w:sig 3</td>
</tr>
<tr>
<td>2 lipwork</td>
<td>227.200</td>
<td>0.000</td>
<td>1016/1516</td>
<td>50 (50)</td>
<td>w:sig 8</td>
</tr>
<tr>
<td>3 iinit</td>
<td>871.000</td>
<td>0.000</td>
<td>1992/2604</td>
<td>100 (100)</td>
<td>w:sem 3</td>
</tr>
<tr>
<td>4 wq:manager</td>
<td>0.000</td>
<td>0.000</td>
<td>368/1252</td>
<td>243 (243)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>80 wq:att_pos_ctl</td>
<td>235324</td>
<td>5.500</td>
<td>4768/6596</td>
<td>244 (244)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>147 dataman</td>
<td>57.000</td>
<td>0.000</td>
<td>752/1188</td>
<td>90 (90)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>20 wq:lp_default</td>
<td>2622.000</td>
<td>3.200</td>
<td>812/1700</td>
<td>205 (205)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>24 wq:i2c1</td>
<td>1782.400</td>
<td>2.100</td>
<td>936/1244</td>
<td>248 (248)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>28 wq:hp_default</td>
<td>3594.100</td>
<td>4.000</td>
<td>1120/1596</td>
<td>243 (243)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>142 wq:spi2</td>
<td>883364</td>
<td>22.200</td>
<td>966/1396</td>
<td>254 (254)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>157 wq:i2c3</td>
<td>1253.000</td>
<td>1.500</td>
<td>824/1244</td>
<td>246 (246)</td>
<td>w:sem 4</td>
</tr>
<tr>
<td>159 wq:spi4</td>
<td>12152.000</td>
<td>3.000</td>
<td>608/1396</td>
<td>251 (251)</td>
<td>w:sem 4</td>
</tr>
</tbody>
</table>

**186 sensors**

- CPU(ms): 96359.200
- CPU(%) : 2.400
- STATE FD : 1096/1964 237 (237)

**Processes:** 24 total, 2 running, 22 sleeping, max FDs: 20

**CPU usage:** 63.00% tasks, 2.00% sched, 44.00% idle

**Memory Usage:** 5120 total, 1024 used 1024 peak

**Uptime:** 3618.000s total, 1900.001s 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 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

Product code generation
Top-level Simulink model of the CLAW for PX4 integration
Simulation & verification environment in MATLAB/Simulink

- The motor/prop model considers effects of motor/arm canted angles ($\phi, \theta, \psi$) 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)
• 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
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
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

[Diagram of PX4 Autopilot system]

[QR code: Youtube link]
Thank you