Master's Thesis

Nonlinear Disturbance Observer Based Path Following for a Small Fixed Wing UAV

Dongmin Shin

Department of Mechanical Engineering

Graduate School of UNIST

2019

Nonlinear Disturbance Observer Based Path Following for a Small Fixed Wing UAV

Dongmin Shin

Department of Mechanical Engineering

Graduate School of UNIST

Nonlinear Disturbance Observer Based Path Following for a Small Fixed Wing UAV

A thesis/dissertation
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Dongmin Shin

6/10/2019

Approved by

Advisor
Hyondong Oh

Nonlinear Disturbance Observer Based Path Following for a Small Fixed Wing UAV

Dongmin Shin

This certifies that the thesis/dissertation of Dongmin Shin is approved.

6/10/2019

signature

Advisor: Hyondong Oh

signature

Hungsun Son

signature

Sanghoon Kang

Abstract

Small fixed-wing UAVs are increasingly attracting attention due to many applications in both military operations such as surveillance, and civilian domains such as powerline patrol and aerial photography. Their popularity growth has reduced the weight and size of small fixed-wing UAVs. They are vulnerable to external disturbances, such as wind due to their reduced size and weight. Disturbance has adverse effects on small fixed-wing UAVs, as it lowers the stability and performance of the control system during the operation. The disturbance includes modeling errors caused by the uncertainty of the system parameters as well as the wind from the external environment.

The disturbance effects acting on small fixed-wing UAVs must be considered explicitly and eliminated eventually. In this regard, various control techniques for compensating the disturbance have been actively studied in control fields. Representative controller design techniques for compensating the disturbance include robust control (RC), sliding mode control (SMC) and adaptive control (AC), and disturbance observer-based control (DOBC). Most of the control techniques cause a slow response in attenuating disturbance effects since feedback control is performed based on tracking errors. Therefore, there is a need to compensate directly for disturbance through feedforward control. The disturbance observer-based control scheme directly estimates the uncertainty of the system, external disturbance, and directly compensates the estimated disturbance through the feedforward.

This paper proposes a nonlinear disturbance observer-based path following controller for a small fixed-wing UAV affected by disturbance such as wind. We used a nonlinear disturbance observer-based control (NDOBC) method for precise path following for a small fixed-wing UAV along with the Lyapunov guidance vector field (LGVF) technique. The disturbance is estimated by a nonlinear disturbance observer, and then it is incorporated into the LGVF path following controller to compensate disturbance effects. The numerical simulation is first carried out through the MATLAB Simulink environment. Software in the loop simulation (SITL) is carried out to verify its performance. Outdoor flight experiments are performed to demonstrate its performance in a real-world environment.

Contents

List of Figures

I Introduction

1.1 Motivation

Small fixed-wing unmanned aerial vehicles (UAVs) are increasingly attracting attention due to many applications in both military operations such as surveillance, and civilian domains such as powerline patrol and aerial photography. Their popularity growth has reduced the weight and size of small fixed-wing UAVs. Among the many types of UAVs, small fixed-wing UAVs (typically less than 10kg) are popular platforms that can be easily deployed, low cost to build, and carry mandatory payloads to perform demanding tasks efficiently. However, they are vulnerable to external disturbances, such as wind due to their reduced size and weight, relatively low airspeed. Disturbance adversely affects flight performance or even causes serious problems on the stability of the control system during the operation. This presents a challenge to the design of the flight control system for small fixed-wing UAVs, here, the disturbance includes modeling errors caused by the uncertainty of the system parameters as well as the wind from the external environment. The disturbance effects on UAVs must be considered and eventually eliminated. The disturbance is difficult to be measured directly using the sensor, and disturbance rejection is one of the main challenges of control system design.

Figure 1: Small fixed-wing unmanned aerial vehicles.

1.2 Research Objectives

Most operations of UAVs essentially consist of commanding UAVs to fly along the path. It is important for the UAVs autonomously to follow a predefined path to perform the mission such as surveillance and search. Most commonly missions of UAVs are to follow either straight-lines or circular-orbit paths. There are many available path following techniques such as Carrot-chasing, nonlinear guidance law (NLGL) in [1], linear quadratic regulator (LQR) in [3], pure pursuit with line-of-sight (PLOS) in [2], and vector field (VF) in [4]. A general requirement of these path following methods is that they must follow the path accurately and robustly in the presence of disturbance such as wind. The detailed analysis for the performance of these path following methods was performed in the varying wind conditions [5]. Carrot-chasing path following method is not accurate and robust to large wind disturbances. The NLGL, PLOS,

and LQR path following methods are sensitive to gains and have a high cross-track error, which means the distance between the path and the UAV. The VF path following method has a low cross-track error and performs the best.

One approaches to remove the effects of wind on path following is to use ground-referenced measurements that are considering wind disturbance. The VF path following method using the direction of flow has demonstrated robustness to wind disturbance by using ground-referenced measurements such as ground speed and course angle instead of airspeed and heading angle [4]. The method using ground-referenced measurements on path following is further investigated by using multiple UAVs [10]. The ground speed and course angle can be provided by the integration of sensors such as inertial navigation system (INS) and global positioning system (GPS). However, in the case of micro UAVs equipped with a low-cost GPS sensor, the quality of the sensor data provided from the GPS may be poor. In addition, any additional measurement provided from the GPS can be very disturbed by gust wind. Instead of using ground-referenced measurements that are considering wind disturbance, another approach to directly estimate and compensate the wind disturbance is needed to remove the wind influence. Similar to the VF path following method, the Lyapunov guidance vector field (LGVF) method for path following is developed in [24]. It was investigated for standoff tracking of UAVs without considering wind disturbance [7-9]. While most of path following methods are used to follow a predefined path using only a single UAV, the LGVF method has the advantage of being able to track a target such as moving ground vehicle using a single or multiple UAVs.

A study of robust control techniques such as adaptive control (AC) and sliding mode control (SMC) has been conducted for compensating for negative effects due to disturbance such as wind, unknown disturbances, and uncertainty. In the case of the robust control techniques, the feedback control technique is basically used and is performed based on the tracking errors between the output state and the desired command. As compared to feedforward control, their methods cause a slow response in attenuating disturbance effects since feedback control is performed based on tracking errors. Therefore, there is a need to compensate directly for disturbance through feedforward control. To perform the feedforward control, it is necessary to measure the disturbance acting on the system. However, the disturbance is difficult to be measured directly using the sensor. Disturbance observer that estimates disturbance by applying the concept of an observer to estimating system disturbance and uncertainty has been studied, disturbance observer-based control scheme that compensates for disturbance by using the estimated disturbance has been studied.

The disturbance observer-based control (DOBC) scheme has two advantages. First, it can be considered as a patch for the designed controller, and it is easy to integrate with the previously designed controller method. Second, the disturbance observer-based control method is an active anti-disturbance control (AADC) method, and it can compensate for the disturbance more quickly than the feedback control based passive anti-disturbance control (PADC) method. As compared with PADC techniques which only reject the disturbances by passive feedback rule,

disturbance observer-based control (DOBC) always leads to a faster dynamic response in dealing with disturbances since a feedforward compensation term is provided in order to directly offset the disturbances in the control systems. Due to the advantages of this DOBC method, DOBC has been regarded as a popular method which estimates and compensates the disturbance. The DOBC technique widely has been used and applied to many industrial systems, robotics, flight control, and astronautic systems [28-30]. Inner loop control with DOBC for UAVs has been investigated in previous studies. It was applied in the attitude controller of a multirotor to reject attitude disturbance [12-15]. In addition, DOBC was applied in the longitudinal controller of a small fixed-wing UAV affected by disturbance such as wind [16], which was further investigated by using DOBC with the linear quadratic regulator (LQR) to improve its performance [17]. However, most of the methods were used in the inner loop to reject the attitude disturbance or be aware of the system model terms of the disturbance observer precisely. Liu et al. [18] proposed the approach to design a guidance controller for a small fixed-wing UAV affected by disturbance such as wind. The DOBC method was applied in the outer loop, where the outer loop behavior of the small fixed-wing UAV was assumed as a kinematic model.

In this paper, we used a nonlinear disturbance observer-based control (NDOBC) method for precise path following for a small fixed-wing UAV along with the Lyapunov guidance vector field (LGVF) technique. The nonlinear disturbance observer is applied to estimate the disturbance, which is then incorporated into the LGVF path following controller to compensate for the disturbance. The NDOBC-based LGVF path following controller as shown in Fig. 2 produces guidance commands such as airspeed, altitude, and heading of UAV to the outer loop controller based on PI control. The PI-based outer loop controller produces attitude commands such as roll and pitch to low-level autopilot based on PIXHAWK, which is commercialized low-cost autopilot hardware with good accessibility. The PIXHAWK is used to stabilize attitude commands in the inner loop.

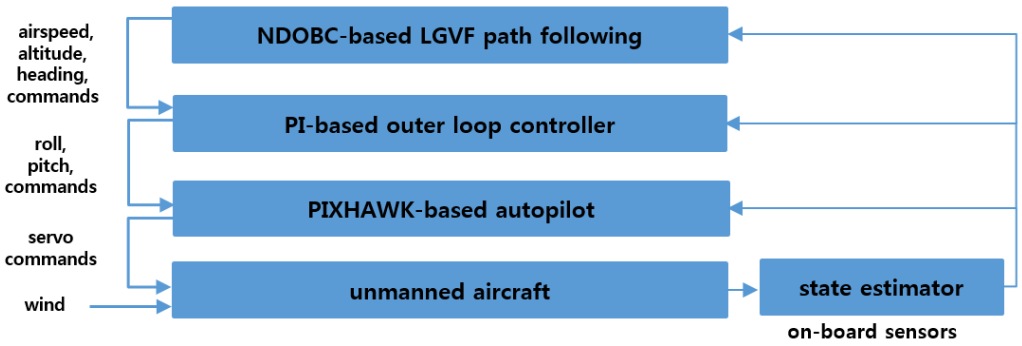

Figure 2: System architecture used in this study.

To develop and test the control algorithm for the UAV, we first perform numerical simulations through the MATLAB Simulink environment. Software in the loop simulation (SITL) is carried out to verify its performance. Outdoor flight experiments using a small fixed-wing UAV are also

performed to demonstrate the performance of the proposed method in a real-world environment.

1.3 Outline of the Thesis

The overall structure of this thesis consists of a total of eight chapters as follows. The motivation and the research objectives are introduced in the first chapter, and a literature survey on researches regarding path following methods and disturbance observer-based control (DOBC) is conducted in chapters 2, 3. The proposed approach used in this study is described in detail in chapter 4. Numerical and software in the loop simulation to verify its performance are described in chapter 5, 6. Outdoor flight experiment results using a small fixed-wing UAV are given and analyzed in chapter 7. Finally, the conclusion of this research and further research topics are discussed in chapter 8.

II Path Following Methods

Path following is generally required for small fixed wing UAVs and any type of unmanned vehicles. Most operations of UAVs essentially consist of commanding UAVs to fly along the path. It is important for the UAVs autonomously to follow a predefined path to perform the mission such as surveillance and search. Most commonly missions of UAVs are to follow either straight-lines or circular-orbit paths.

There are many available path following methods, it is important for UAVs to follow the path precisely and robustly in order to carry out its mission. The common and widely used path following methods are compared based on tracking accuracy, and robustness to wind disturbance. The methods compared are Carrot-chasing path following, nonlinear guidance law (NLGL) in [1], linear quadratic regulator (LQR) in [3], pure pursuit with line-of-sight (PLOS) in [2], and vector field (VF) in [4].

2.1 Path Following Problem

The path following problem is to decide the desired heading angle of the UAV. The UAV tracks the desired heading angle which is provided from path following methods. In Figure 3, the heading angle and initial position of the UAV are ψ and $p(x, y)$. Most commonly missions of UAVs are to follow either straight-lines paths or circular-orbit paths [5]. Figure 3(a) shows the geometry of the straight-lines. The straight-lines paths can be defined by using waypoints which can be followed by the UAV. The line-of-sight (LOS) is the angle for the x-y coordinate frame of this straight-line. Figure 3(b) shows the circular-orbit which is called the loiter path following. The distance d is the cross-track error that must be minimized. It means the distance between

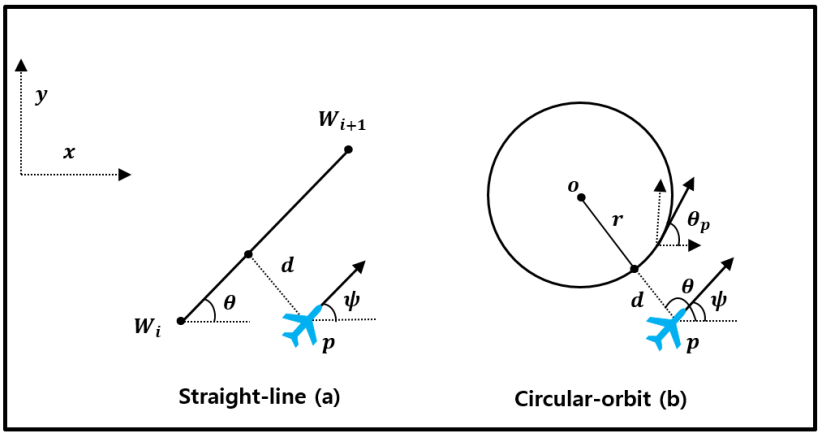

Figure 3: Path following geometry for the straight-line and circular-orbit.

the UAV to the path. θ_d is defined as the heading angle of the path. The UAV must minimize heading errors as $|\theta_d - \psi|$. The path following methods aim to have $d \to 0$ and $|\psi - \theta| \to 0$, as $t \to \infty$.

2.2 Carrot-Chasing

Carrot-chasing path following method uses a virtual target point (VTP). The VTP is a virtual target point on the path. Carrot-chasing path following method makes the heading angle updated toward the virtual target point. The UAV tracks the heading angle decided by Carrot-chasing path following method. As time elapses, the UAV moves along the path and results in following the path. In the missile guidance and aerospace systems, the Carrot-chasing path following method is investigated. But, it is not robust and accuracy for disturbance such as wind. The UAV takes a long time to reach the VTP, has a high cross-track error.

2.3 Pure Pursuit and LOS

The pure pursuit and LOS-based path following (PLOS) combines the pure pursuit and LOS-based guidance law. The PLOS method was mainly applied to the missile guidance systems in aerial fields [1] and [2]. The UAV moves to the waypoint W_{i+1} by the pure pursuit guidance law. At this time, the LOS guidance method leads the UAV to the LOS. It makes the LOS angle equal to the angle between the waypoint and the UAV. The PLOS gains must appropriately be chosen. The PLOS path following method is sensitive to the PLOS gains and cannot follow the path when wind disturbance is high.

2.4 Linear Quadratic Regulator

While most of the path following techniques calculate the desired heading angle by using the geometry of the path. The goal of the linear quadratic regulator (LQR) path following is to decide the minimal input which is calculated using optimal control theory. The optimal control theory makes it possible to decide the input to minimize the cost function. The minimal input is the control effort calculated to minimize the cross-track error. But it has a high cross-track error, and the UAV cannot follow the path when wind disturbance is high. The detail of the LQR path following is described in [3].

2.5 Nonlinear Guidance Law

The nonlinear guidance law (NLGL) path following method also uses the concept of the virtual target point (VTP). It draws one circle of radius assuming that the UAV tracks the path. At the current position, the path can be intercepted by the circle of the radius at two points. Two points of the path are selected according to the direction in which the UAV must travel. This method has the advantage that the cross-track error is lower than the above path following methods such as Carrot-chasing, PLOS, and LQR.

2.6 Vector Field

The vector field (VF) path following method uses the direction of flow which is represented by the vector field. If the heading of the UAV is calculated according to the vector field direction, then the UAV can follow the path. Lyapunov stability arguments show the stability of the vector field approach in [4]. The VF path following approach has a low cross-track error compared with the NLGL method and any type of methods [5].

The vector field path following method has demonstrated robustness to wind disturbance [4]. Instead of heading angle and airspeed, ground-referenced measurements such as course angle and ground speed were used to compensate for wind disturbance. The ground speed and course angle can be provided by the integration of sensors such as inertial navigation system (INS) and global positioning system (GPS). But, it purely relies on UAV's ground track [4]. In the case of micro UAVs equipped with a low-cost GPS sensor, the quality of the sensor data provided from the GPS may be poor. In addition, any additional measurement provided from the GPS can be very disturbed by gust wind. Instead of using ground-referenced measurements that are considering wind disturbance, another approach to directly estimate and compensate the wind disturbance is needed to remove the wind influence.

Similar to the VF path following method, the Lyapunov guidance vector field (LGVF) method for path following is developed in [24]. It was investigated for standoff tracking of UAVs without considering wind disturbance [7-9]. While most of path following methods are used to follow a predefined path using only a single UAV, the LGVF method has the advantage of being able to track a target such as moving ground vehicle using a single or multiple UAVs.

III Disturbance Observer Based Control Methods

Disturbances widely exist in control systems and adversely affects the performance or stability of the control systems. The disturbance as shown in Fig. 4 includes modeling errors caused by the uncertainty as well as the wind from the external environment. The disturbance is difficult to be measured directly using the sensor, and disturbance rejection is one of the main challenges of control system design.

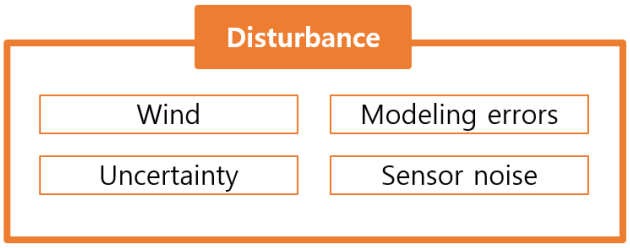

Figure 4: Disturbance affecting flight performance of the control systems.

Disturbance observer-based control technique has been regarded as a popular method which estimates and compensates the disturbance. DOBC achieves a better nominal dynamic performance since the nominal performance is restored if there is no disturbance or uncertainty. In the case of the existing robust control techniques, the feedback control technique is basically used to reject disturbances. Disturbance rejection is performed based on the errors between the output state and the desired command [16]. As compared to feedforward control, their methods cause a slow response in attenuating disturbance effects since feedback control is performed based on tracking errors. Therefore, there is a need to compensate directly for disturbance through feedforward control. To perform the feedforward control, it is necessary to measure the disturbance acting on the system. However, the disturbance is difficult to be measured directly using the sensor. Disturbance observer that estimates disturbance by applying the concept of an observer to estimating system disturbance and uncertainty has been studied, disturbance observer-based control scheme that compensates for disturbance by using the estimated disturbance has been studied.

3.1 Basic Framework

Figure 5 shows a basic framework of disturbance observer-based control technique. The composite parts are the feedback and the feedforward controllers. The feedback controller is generally used to track errors. At this step, the disturbance and uncertainty of the control system need not be considered. The disturbance observer estimates the disturbance and uncertainty of control systems, and then the feedforward controller compensates for the estimated disturbance.

The disturbance observer estimates and compensates uncertainty, and disturbance such as wind. DOBC has many advantages due to its active anti-disturbance control (AADC) charac-

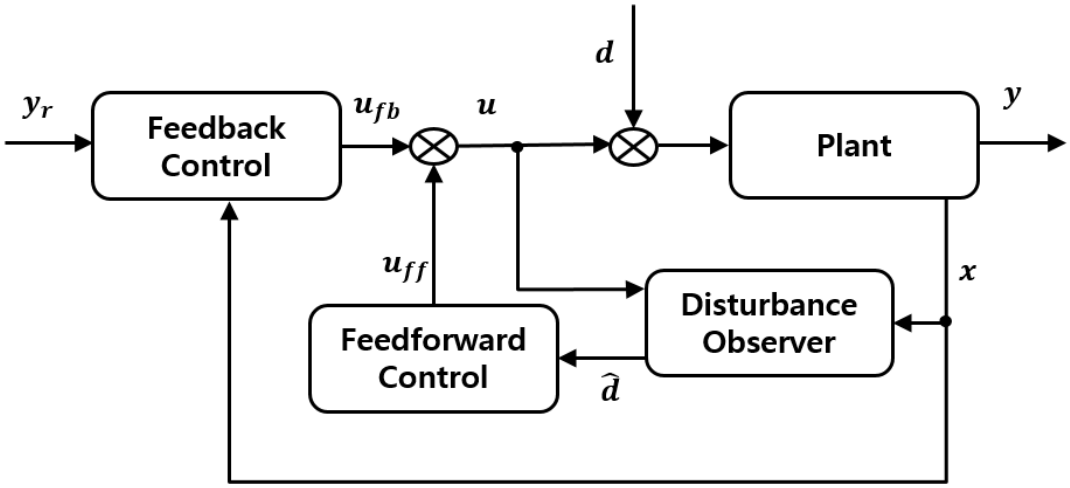

Figure 5: A basic framework of disturbance observer based control.

teristics. As compared with PADC techniques which only reject the disturbances by passive feedback rule, disturbance observer-based control (DOBC) always leads to a faster dynamic response in dealing with disturbances since a feedforward compensation term is provided in order to directly offset the disturbances [28]. DOBC achieves the tracking and disturbance rejection performance by using the feedback and feedforward controllers. These promising features have resulted in several advantages as compared with the PADC methods.

DOBC method can be considered as a patch for the designed controller. It is easy to integrate with the previously designed controller. This has the advantage that there is no change in the designed controller. It means that DOBC is patched to the designed controller in order to improve disturbance rejection abilities and the robustness. Therefore, it can achieve both tracking and disturbance rejection performances. The DOBC technique widely has been used and applied to many industrial systems, robotics, flight control, and astronautic systems [28-30].

3.2 Frequency Domain Disturbance Observer

The frequency domain disturbance observer is shown in Fig. 6. The disturbance observer using frequency domain was mainly applied to the attitude controller to reject the attitude disturbance [12-15]. ϕ_d, ψ_d, and θ_d are the desired roll, heading, and pitch. u_p and d are input and disturbance. $P(s)$ is the actual system with disturbances such as modeling errors caused by the uncertainty as well as the wind from the external environment. $C(s)$ is the attitude controller which stabilizes the inner loop. It is designed for the nominal system $P_n(s)$ to track the desired command such as ϕ_d, θ_d, and ψ_d without considering disturbances.

The single-input-single-output (SISO) system can be expressed as:

$$\dot{x} = f(x) + gu + gd \tag{1}$$

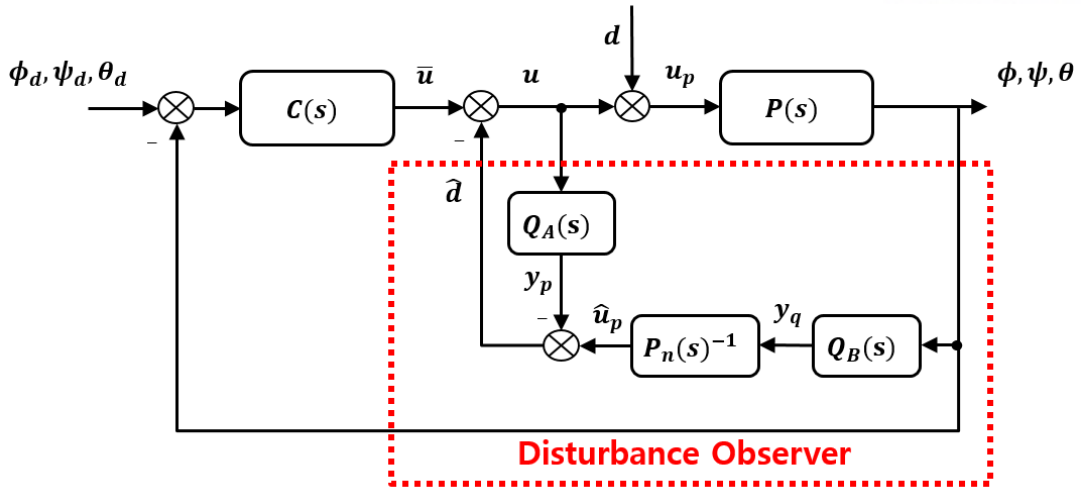

Figure 6: A block diagram of the frequency domain disturbance observer.

The behavior of Eq. (1) is unexpected since the actual system $P(s)$ is uncertain. Thus, it is required to design the controller robustly. The nominal system $P_n(s)$ can be defined as:

$$\dot{x} = f_n(x) + g_n \bar{u} \tag{2}$$

where $\bar{u}$ is the external input from the attitude controller. The desired control input can be decided so that the system of Eq. (1) with disturbances becomes equivalent to the nominal system of Eq. (2) without disturbances as:

$$u_{desired} = -d + \frac{1}{g}\{-f(x) + f_n(x)\} + \frac{g_n}{g}\bar{u} \tag{3}$$

However, using the desired control input $u_{desired}$ is not directly feasible since the $f(x), g$, and d of actual system are unknown. The frequency domain disturbance observer aims to calculate the attitude controller output y_p, and the system input u_p. The system input u_p includes the disturbances, and can be computed using inverse of the nominal model $P_n^{-1}(s)$ as:

$$u = \bar{u} + y_p - \hat{u}_p = \bar{u} - \hat{d} \tag{4}$$

The estimated disturbance $\hat{d}$ can be represented from Fig. 6 as follows:

$$\hat{d} = Q_B(s)P_n(s)^{-1}y - Q_A(s)u \tag{5}$$

where $Q_A(s)$ and $Q_B(s)$ are Q-filter. The Q-filters taken as stable low pass filters should have relative degrees more than that of the nominal plant as:

$$Q_A(s) = Q_B(s) = \frac{a_0/\tau^2}{s^2 + (a_0/\tau)s + a_0/\tau^2} \tag{6}$$

where a_0 to make unity DC gain, τ is a positive real constant.

3.3 Time Domain Disturbance Observer

The time domain disturbance observer is shown in Fig. 7. The time domain disturbance observer is applied in the longitudinal controller of a small fixed-wing UAV affected by disturbance such as wind [16], which was further investigated by using the time domain disturbance observer with the linear quadratic regulator (LQR) to improve its performance [17].

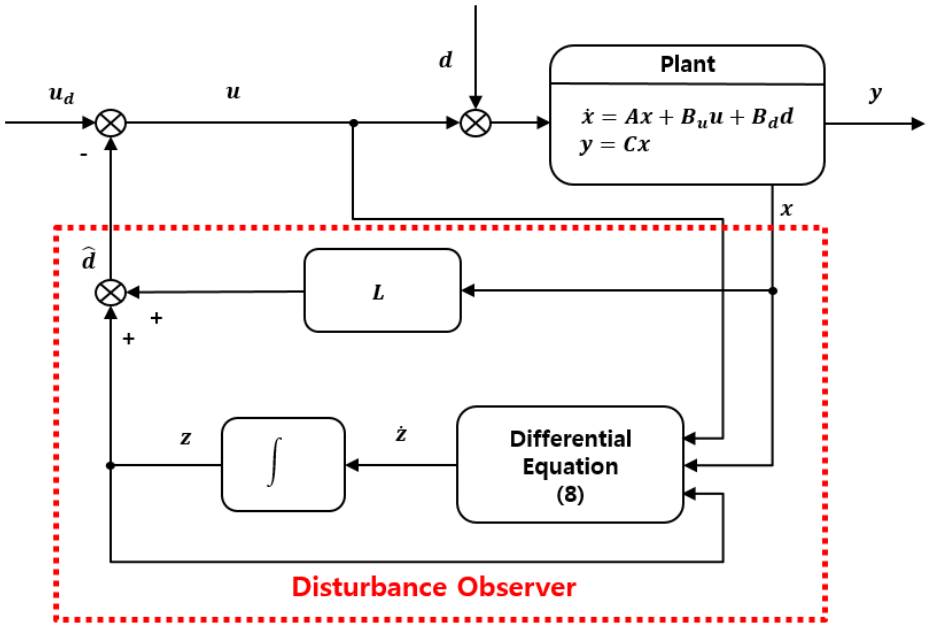

Figure 7: A block diagram of the time domain disturbance observer.

The multi-input-multi-output(MIMO) system can be expressed as follows:

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_d d \\ y &= Cx \end{aligned} \tag{7}$$

where $x \in R^n$, $d \in R^r$, $y \in R^l$, and $u \in R^m$ are state, disturbance, the output, and the input. A, B_d, B_u, and C are system matrices.

The time domain disturbance observer aims to estimate and compensate for the disturbance in the control system Eq. (7), given by [19].

$$\begin{aligned} \dot{z} &= -LB_d(z + Lx) - L(Ax + B_u u) \\ \hat{d} &= z + Lx \end{aligned} \tag{8}$$

where $\hat{d}$, L, and z are estimated disturbance, the observer gains to be tuned for performance, and the internal state. To prove the ability of the disturbance observer, the disturbance estimation error between the true disturbance and estimated disturbance can be expressed as:

$$e_d = d - \hat{d} \tag{9}$$

Taking the derivative of Eq. (9) along with Eq. (8) and Eq. (7) gives as:

$$\dot{e}_d = -LB_d e_d - \dot{d} \tag{10}$$

Provided that the disturbances tend to constant meaning that disturbances vary slowly, and the gain L is appropriately determined such that $-LB_d$ is Hurwitz, the Eq. (10) is shown that disturbance estimation error is asymptotically stable.

3.4 Nonlinear Disturbance Observer

Nonlinear disturbance observer-based control (NDOBC) method was first proposed by Chen et al in [20] to estimate and compensate disturbance such as unknown friction in the manipulator's systems. It was further investigated to address uncertainty and modeling error in nonlinear flight dynamics of missiles for autopilot controller design. The flight performance has improved a lot [21]. Liu et al proposed the method to design the guidance controller for a small fixed-wing UAV affected by disturbance such as wind. The NDOBC method was applied in the path following controller, where the outer loop behavior of the UAV was assumed as a kinematic model [18].

A class of affine nonlinear system can be expressed as follows:

$$\begin{aligned} \dot{x} &= f(x) + g_1(x)u + g_2(x)d \\ y &= h(x) \end{aligned} \tag{11}$$

where $x \in R^n$, $d \in R^q$, $y \in R^s$, and $u \in R^m$ represent the state, the disturbance, output, and input. NDOBC was proposed in [20] and [23] as:

$$\begin{aligned} \dot{z} &= -l(x)g_2(x)z - l(x)[g_2(x)p(x) + f(x) + g_1(x)u] \\ \hat{d} &= z + p(x) \end{aligned} \tag{12}$$

where $z \in R^q$ is the internal state of the nonlinear disturbance observer (NDO), the nonlinear function $p(x)$ is to be designed. The NDO gain $l(x)$ can be decided by:

$$l(x) = \frac{\partial p(x)}{\partial x} \tag{13}$$

It has been shown in [20] that the nonlinear disturbance observer asymptotically estimates the disturbance, provided that the observer gain $l(x)$ is chosen such as:

$$\dot{e}_d = -l(x)g_2(x)e_d \tag{14}$$

where $e_d = d - \hat{d}$ is the disturbance estimation error. The Eq. (14) is asymptotically stable about appropriately chosen gain $l(x)$ regardless of x. The NDO gain $l(x)$ can be chosen by many ways, which was suggested by [22] and [23].

IV Approach

The outer loop controller including heading, altitude, and speed controller is designed as a proportional-integral (PI) control method with anti-windup which works to relieve the accumulation of integrators, and it makes possible for the UAV to follow the commanded heading, altitude, and speed.

Lyapunov guidance vector field (LGVF) path following technique which was mainly investigated for standoff tracking of UAVs without considering wind disturbance in [7-9] is used to follow the reference circular path for the UAV. The nonlinear disturbance observer-based control (NDOBC) method which used by Liu et al in [18] is applied for precise path following control for the UAV affected by disturbance such as wind. The disturbance estimated by the nonlinear disturbance observer (NDO) is incorporated into the LGVF path following controller to compensate disturbance influences.

4.1 Outer Loop Controller Design

Proportional-integral-derivative (PID) control is used to control the outer loop. PID is a classical control method widely used in the fields of flight control and as well as industrial control. It became available to design and analyze the system since linear system theory was developed in the 1960s. It is simple to design and implement since the design of the PID control method is response-based. Therefore, it can be easily designed and tuned without knowledge of the system model.

Figure 8 shows the outer loop controller scheme used in this study. The outer loop controller is designed as PI control with anti-windup. ψ_{cmd} is the commanded heading from path following controller. h_{cmd} and V_{cmd} are the commanded altitude and commanded speed of the UAV, respectively.

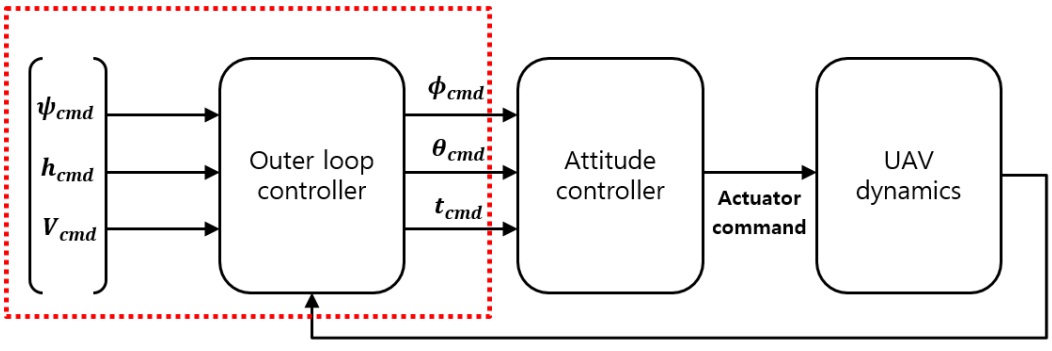

Figure 8: Outer loop controller scheme.

$$
\begin{aligned}
e_\psi &= (\psi_{cmd} - \psi) \\
e_h &= (h_{cmd} - h) \\
e_V &= (V_{cmd} - V)
\end{aligned} \tag{15}
$$

where e_ψ, e_h, and e_V are the error of heading, altitude, and speed, respectively.

$$
\begin{aligned}
\phi_{cmd} &= K_p^\psi * e_\psi + K_i^\psi \int e_\psi d\tau \\
\theta_{cmd} &= K_p^h * e_h + K_i^h \int e_h d\tau \\
t_{cmd} &= K_p^V * e_V + K_i^V \int e_V d\tau
\end{aligned} \tag{16}
$$

where $K_{p,i}^{\psi,h,V}$ represents proportional and integral gains of outer loop controller. ϕ_{cmd}, θ_{cmd}, and t_{cmd} is commanded roll, commanded pitch, and trust which is caused from outer loop with PI control.

In this study, the anti-wind up as shown in Fig. 9 is applied to relieve the accumulation of integrators in the outer loop controller with PI control. Integrator anti-wind up happens when the error form Eq. (15) persists. Anti-wind up schemes are intended to limit the integrator from winding up after u such as ϕ_{cmd}, θ_{cmd}, and t_{cmd} is saturation.

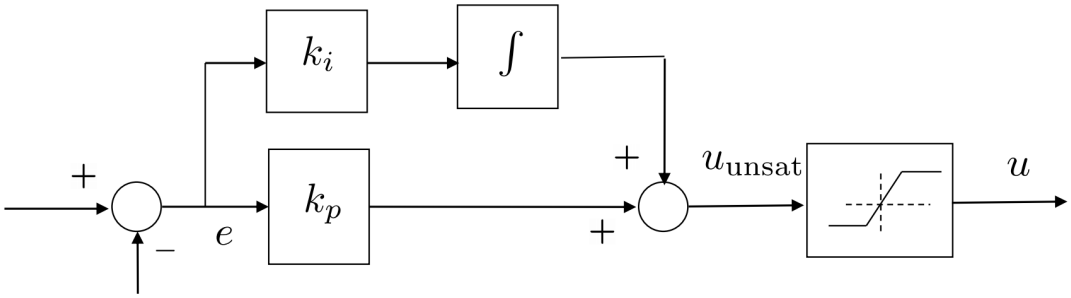

Figure 9: Integrator anti-wind up scheme.

Let the control before the anti-wind up update be given as:

$$
u_{unsat}^- = K_p e + K_i I^- \tag{17}
$$

where $I = \int e d\tau$ is integrator. The control after the anti-wind up update be given as:

$$
u_{unsat}^+ = K_p e + K_i I^+ \tag{18}
$$

Subtracting the two gives as:

$$
u_{unsat}^+ - u_{unsat}^- = K_i (I^+ - I^-) \tag{19}
$$

$$I^+ = I^- + \frac{1}{K_i}(u^+_{unsat} - u^-_{unsat}) \tag{20}$$

where u^+_{unsat} is selected to be the saturation limit. Anti-wind up is applied when u^-_{unsat} is more than the saturation limit.

4.2 Unmanned Aerial Vehicle Kinematics

Provided that a UAV has a low-level autopilot system for heading, speed, and altitude hold function, this study aims to design guidance command inputs to the low-level autopilot system for path following. With separation principles, assuming that the bandwidth of the low-level flight autopilot system is 5 to 10 times faster than the bandwidth of the path following controller, they can be designed separately.

The two-dimensional kinematics equation of the UAV is used for path following problem as follow:

$$\begin{aligned} \dot{x} &= V_a cos\psi + W_x \\ \dot{y} &= V_a sin\psi + W_y \\ \dot{\psi} &= u \end{aligned} \tag{21}$$

where V_a, ψ, and u are the speed, heading, and yaw rate of the UAV, respectively. W is the wind disturbance.

4.3 Nonlinear Disturbance Observer Design

To facilitate the application of the nonlinear disturbance observer (NDO), Eq. (21) is converted into a general form as:

$$\dot{x} = f(x) + g_1(x)u + g_2(x)d \tag{22}$$

where functions $f(x)$, $g_1(x)$, and $g_2(x)$ can be easily derived from Eq. (21). Provided that the disturbance is nearly constant, that is, $\dot{d} \approx 0$, the nonlinear disturbance observer used to estimate and compensate the disturbance is given as:

$$\begin{aligned} \dot{z} &= -l(x)g_2(x)z - l(x)[g_2(x)p(x) + f(x) + g_1(x)u] \\ \hat{d} &= z + p(x) \end{aligned} \tag{23}$$

where $\hat{d} = [\hat{W}_x \ \hat{W}_y]^T$ is the wind disturbance estimated by the NDO, $p(x)$ is the nonlinear function to be designed, z is the internal state of the NDO, and $l(x)$ is the NDO gain given by

$$l(x) = \frac{\partial p(x)}{\partial x} \tag{24}$$

$e = d - \hat{d} = [e_x \ e_y]^T$ is the estimation error of the above-mentioned NDO. Assuming that the disturbances tend to constant and vary slowly compared to the observer dynamics, and by

combining Eq. (22), (23), and Eq. (24), the estimation error can be expressed as following properties:

$$\dot{e} = \dot{d} - \dot{\hat{d}} = -\dot{z} - \frac{\partial p(x)}{\partial x}\dot{x} = -l(x)g_2(x)e \tag{25}$$

Therefore, the disturbance observer design problem can be dealt with as a problem of selecting the NDO gain $l(x)$. If the NDO gain $l(x)$ is appropriately chosen, Eq. (25) can be asymptotically stable regardless of the state x. The $g_2(x)$ is a constant matrix, the NDO gain can be selected as:

$$l(x) = L = \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \tag{26}$$

where l_x and l_y are gains to be tuned. The NDO gains decide the converging rate of the estimation error.

4.4 Lyapunov Guidance Vector Field

The Lyapunov guidance vector field (LGVF) technique is used to follow the reference circular path for the UAV. The Lyapunov function can be expressed as:

$$V(x, y) = (r^2 - r_d^2)^2 \tag{27}$$

where $r = \sqrt{\delta x^2 + \delta y^2}$ is the relative distance of the UAV. r_d is a desired distance of the UAV. The total time derivative of V is given by differentiating Eq. (27) as:

$$\dot{V}(x, y) = \nabla V[\dot{x}, \dot{y}]^T \tag{28}$$

Eq. (28) is can be negative by selecting desired velocity of the UAV according to the guidance vector field. The LGVF uses the following desired velocity $[\dot{x}_d, \dot{y}_d]^T$ as:

$$\begin{bmatrix} \dot{x_d} \\ \dot{y_d} \end{bmatrix} = \frac{-v_d}{k_l r(r^2 + r_d^2)} \begin{bmatrix} \delta x(r^2 - r_d^2) + \delta y(2rr_d) \\ \delta y(r^2 - r_d^2) - \delta x(2rr_d) \end{bmatrix} \tag{29}$$

where v_d is a desired speed of UAV. k_l is positive constant which newly introduced in [27], is gain which can adjust the converging rate of UAV to the circular path. The desired heading as shown in Fig. 10 can be determined as:

$$\psi_d = tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right) \tag{30}$$

where it uses desired two-dimensional velocities given by the vector field in Eq. (29). The desired velocity is used to produce the guidance command for turn rate. The guidance command u_w for turn rate is chosen as the sum of proportional feedback and feedforward terms as:

$$u_w = -k_w(\psi - \psi_d) + \dot{\psi}_d \tag{31}$$

$$\dot{\psi}_d = 4v_d \frac{r_d r^2}{(r^2 + r_d^2)^2} \tag{32}$$

where $\dot{\psi}_d$ is desired turning rate which can be obtained by differentiating Eq. (30).

The wind disturbance is estimated by nonlinear disturbance observer (NDO) in Eq. (23) and the target is fixed, the guidance vector is adjusted to consider the velocity $[T_x, T_y] = [W_x, W_y]$. To compensate for the disturbance estimated from the nonlinear disturbance observer, the new desired velocity of LGVF used for Eq. (29) can be computed as:

$$\begin{bmatrix} \dot{x}_{dn} \\ \dot{y}_{dn} \end{bmatrix} = \begin{bmatrix} \hat{W}_x + \alpha_s \dot{x}_d \\ \hat{W}_y + \alpha_s \dot{y}_d \end{bmatrix} \tag{33}$$

where $\hat{W}_x, \hat{W}_y$ are the disturbance estimated from Eq. (23), and α_s is a scale factor which can be obtained by equating the magnitude of the original and new desired velocity. The final revised desired heading can be obtained as follow:

$$\psi_d = tan^{-1}\left(\frac{\dot{y}_{dn}}{\dot{x}_{dn}}\right) \tag{34}$$

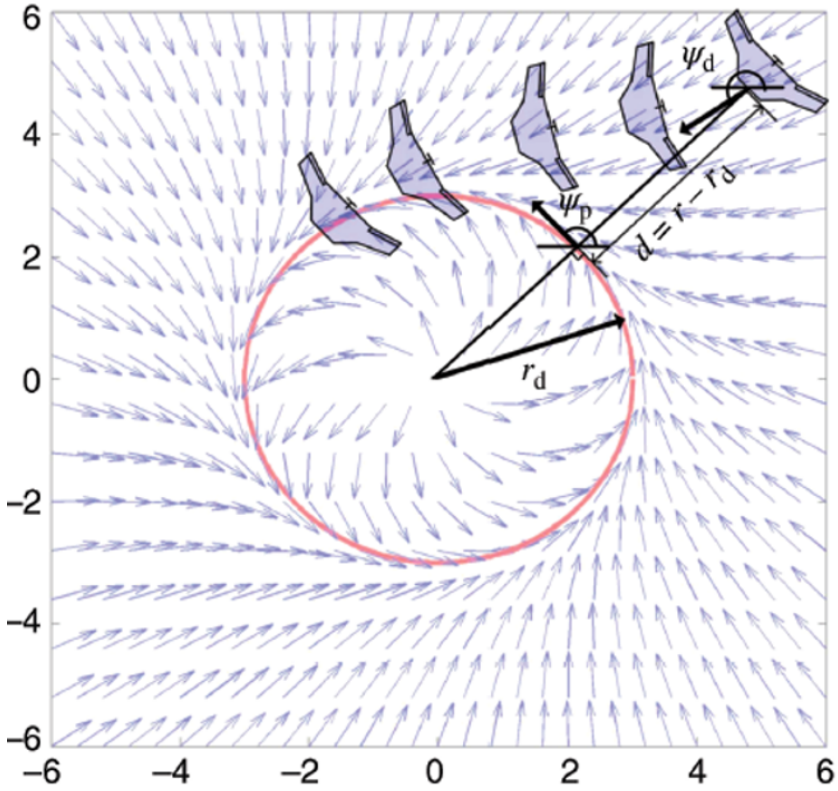

Figure 10: The geometry of tangent vector field [8].

4.5 The Proposed NDOBC-Based LGVF Controller

Figure 11 shows the nonlinear disturbance observer-based LGVF path following controller structure used in this study. The disturbance information such as wind, uncertainty is estimated by the nonlinear disturbance observer (NDO), then it is incorporated into the LGVF path following controller to compensate disturbance influences. The commanded heading from the LGVF controller, speed, and altitude are controlled by the outer loop controller with PI control and anti-windup. It is assumed that the UAV has a low-level autopilot system that stabilizes the inner loop.

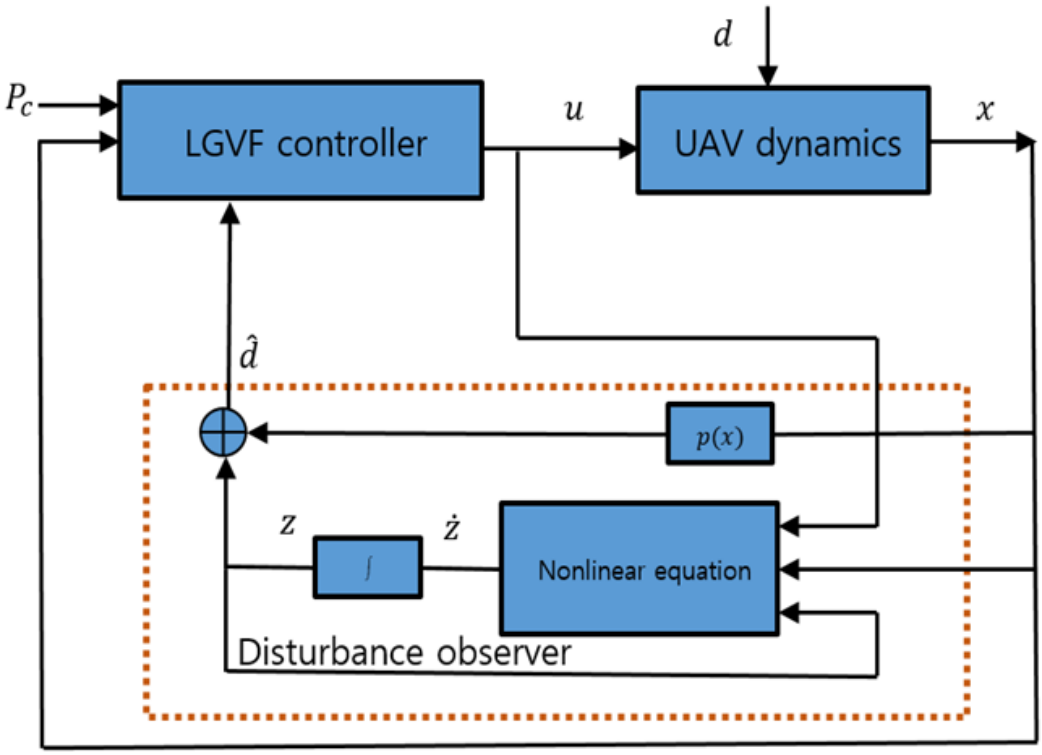

Figure 11: Controller structure.

V Numerical Simulation

Numerical simulation is first carried out through the MATLAB Simulink environment for the proof of concept. MATLAB Simulink simulation using the LGVF and DOBC methods is carried out using a two-dimensional kinematic model of the UAV instead of using a controller such as PI controller in order to easily verify the performance of the DOBC.

5.1 MATLAB Simulink Structure

Figure 12 shows the MATLAB Simulink structure used to verify the performance of the DOBC method. In the Simulink environment, the wind disturbance is set at 10 m/s in the y-direction. The MATLAB Simulink algorithm includes the nonlinear disturbance observer which estimates the disturbance such as wind, LGVF path following controller which compensates the estimated disturbance in order to follow the precise path of the UAV affected by disturbance such as wind.

Table 1: Parameter in MATLAB Numerical Simulation.

Parameter	Value
l_x, l_y	0.2
k_l	0.3
k_w	2
r_d	100
v_d	25

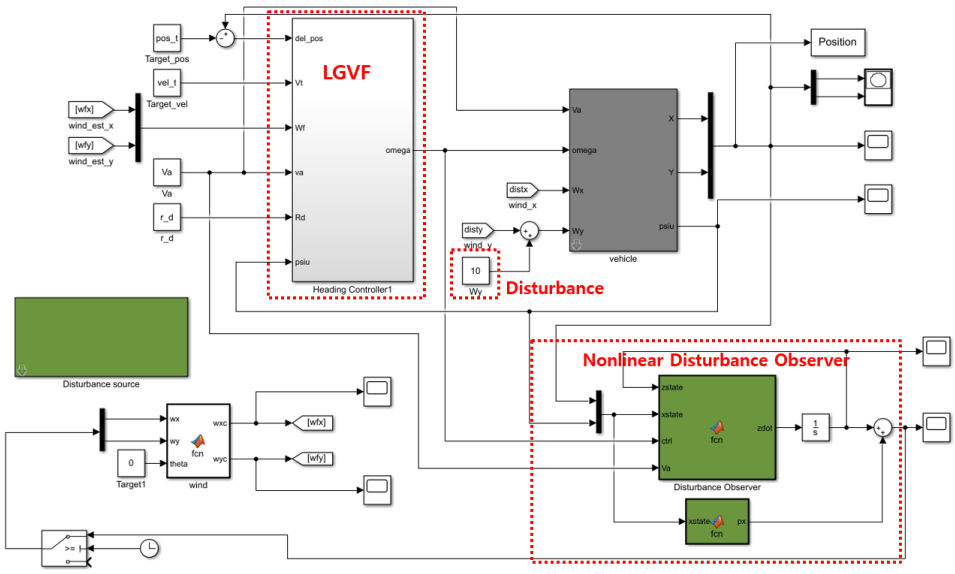

Figure 12: MATLAB Simulink structure.

5.2 Simulation Result

Figure 13 shows the MATLAB Simulink simulation result. The UAV follows a reference circular path in an environment with a wind speed of 10 m/s. The speed of UAV is 25m/s, The DOBC is turned on after 10 seconds. Initially, the UAV follows the path far away from the reference path without DOBC. Once the DOBC is turned on, the UAV follows the reference path accurately. Figure 14 shows that the nonlinear disturbance observer estimates reference wind disturbance of 10m/s in y-direction accurately.

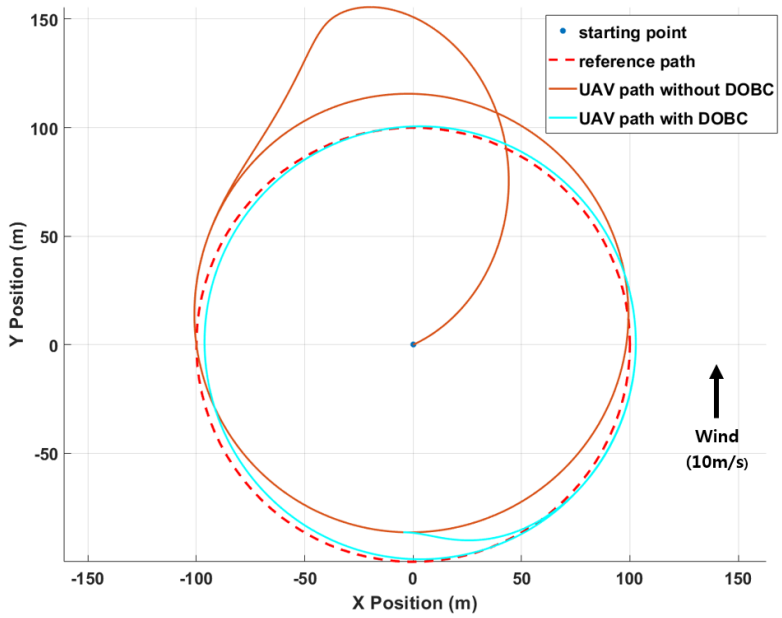

Figure 13: MATLAB Simulink simulation.

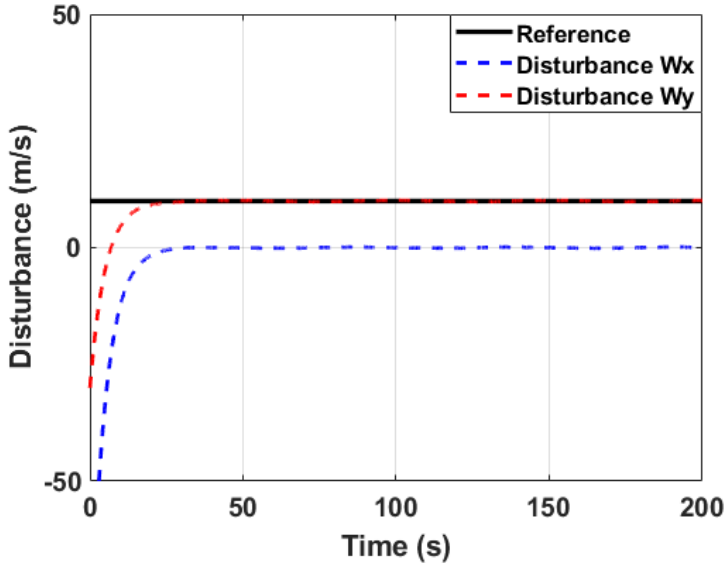

Figure 14: Disturbance estimated by nonlinear disturbance observer.

VI Software In The Loop Simulation

In this study, software in the loop simulation using Gazebo simulator with PX4 open source autopilot software and robot operating system (ROS) is carried out. The software in the loop simulation enables the development of algorithms that can be deployed in actual UAVs with many cases.

6.1 Gazebo Simulator

The gazebo is an open-source robot simulator. Since it is developed by Nate Koenig in 2002, it has been used by many robot developers to date and is now well known as a representative project of the Open-source robot foundation (OSRF) with ROS. Gazebo which is a 3D dynamic simulator can accurately simulate robots such as an unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) in outdoor and indoor environments. Gazebo which includes physics engines provides physics simulations of vehicles, sensor suites. It is possible to easily test and debug robotics algorithms with realistic scenarios by using Gazebo. Furthermore, the Gazebo simulator provides excellent scalability through plugins that allow a user to access and fast interface with other systems such as software-in-the-loop simulation with PX4 or ROS (Robot Operating System). Figure 15 shows a fixed-wing aircraft model of a PX4-based Gazebo simulator.

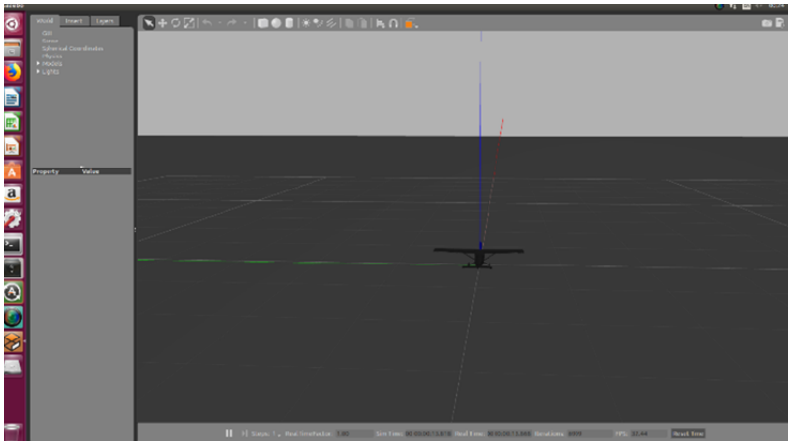

Figure 15: Fixed-wing aircraft model of a PX4-based Gazebo simulator.

6.2 PX4 Autopilot Software

PX4 is an open-source autopilot software for UGV and UAV. The PX4 offers many flexible tools for many robot developers to share their skills and develop customized solutions for missions of UGV and UAV. PX4 has grown since the PIXHAWK project started at ETH Zurich, where the flight control system was initially designed based on computer vision. The PIXHAWK project is currently contributed by more than 300 global developers and is being contributed by the developers of most innovative companies in the world to apply UAVs in the unmanned aerospace

industry fields. There is an open-source community about the PX4 software and PIXHAWK hardware in the field of unmanned aerial vehicles today. PX4 which belongs to drone code has a nonprofit organization managed by the Linux Foundation to facilitate using the open-source software of UAVs.

PX4 consists of two-part which is called the flight stack and the middleware. The flight stack is the software of the flight control system and estimation, and the middleware is a robot part that can support all types of UAVs and UGVs, offering communication and hardware integration. Figure 16 shows the software architecture of the PX4.

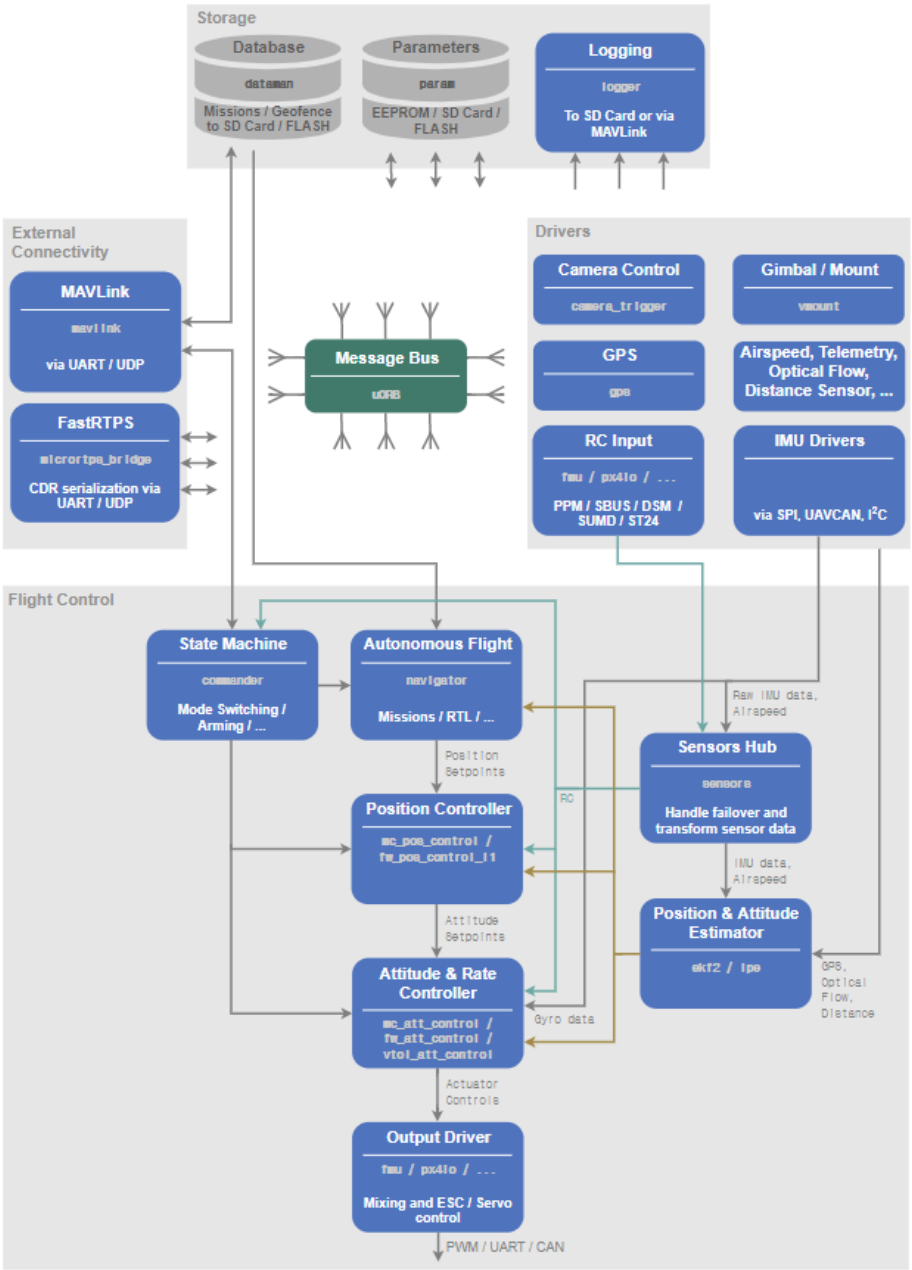

Figure 16: Software architecture of PX4 [31].

The top part of the diagram contains the middleware block and the bottom is the part of

the flight stack. The source code is divided into standalone modules and programs marked as monospace in the figure. In general, each building block is a module. The arrows indicate the flow of information and connections between the building blocks which corresponds to each module. In practice, there are more connections than marked, and most modules access some message data, such as parameters. Modules can communicate with each other via called uORB which can publish and subscribe to some message data. The system is reactive, asynchronous and updates immediately as new data becomes available. The system fully parallelizes communications and all operations.

The PX4 flight stack includes software of navigation, guidance, and control for UAVs and UGVs. It has various controllers and estimators. The controllers are for fixed-wing, VTOL, and multirotor airframes. The estimators are for position and attitude. Figure 17 shows a diagram of the PX4 flight stack. It includes the sensors, RC remote input and navigator, and down to the motor or surface control such as actuators. One or more sensor data can be taken, combined

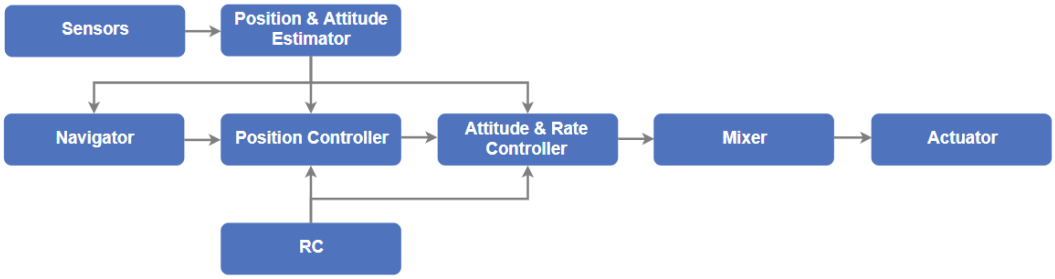

Figure 17: Diagram of the PX4 flight stack [31].

by an estimator that estimates the state of a vehicle. For example, the attitude of a vehicle is estimated by using the IMU sensor data. A controller uses a setpoint which is the desired command, output state which is measured by the sensor and estimated by estimator. It aims to adjust the value of the output state to match the desired command. The output state eventually reaches that desired command by the feedback controller. For example, the position controller uses the desired position as the command, the output state is the currently estimated position by the estimator, and the output is the attitude and thrust command that makes the vehicle move to the desired position by attitude controller. A mixer mixes force commands which are from the attitude controller and divided into individual motor inputs.

Figure 18 shows a fixed-wing inner loop controller in the PX4 flight stack. The attitude controller of Fig. 18 is used to stabilize the inner loop in this study. The attitude controller of a UAV uses a cascaded loop method. The outer loop calculates the error between the desired attitude and the estimated attitude. It uses the proportional (P) controller which generates a rate command. Then the inner loop calculates the rate command error and uses the proportional-integral (PI) controller which generates the commanded angular acceleration. The angular position of the control surfaces such as rudder, elevator, and aileron are then calculated using prior knowledge of the system and the commanded angular acceleration through the control

mixing. The attitude controller is tuned for the airspeed measurements which can be measured by the airspeed sensor due to effect at high speed. The aircraft's body axis force and moments are created by the control surfaces and aerodynamic damping interfering with motion proportional to body rates. The damping can be offset by using feedforward in the rate loop in order to maintain a constant speed. The attitude controllers such as pitch and roll have the same structure. It is assumed that the longitudinal and lateral dynamics models are decoupled.

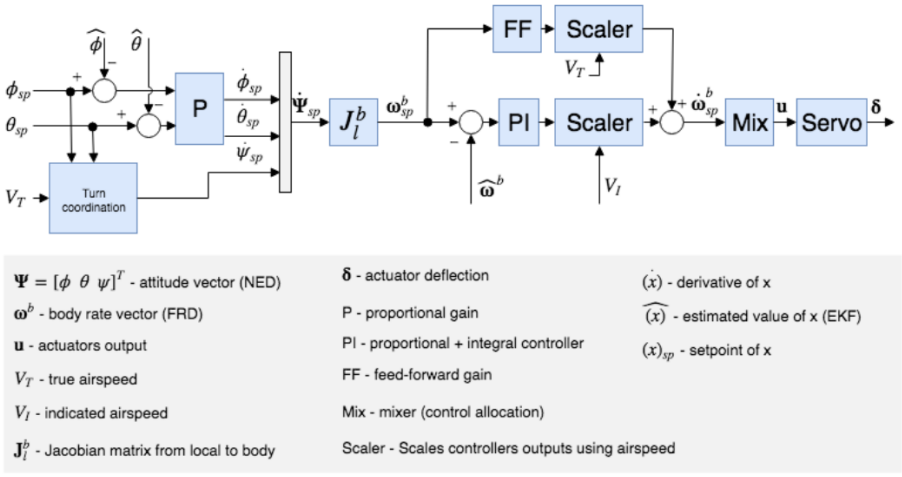

Figure 18: PX4 Fixed-Wing attitude controller used in this study [31].

The PX4 middleware includes drivers of system devices for sensor modules, communication with the external environment (on-board computer, ground control station) and the uORB which is PX4 internal communication mechanisms, which publish and subscribe messages. The middleware allows PX4 autopilot software to be executed on the desktop operating system which can use ROS (Robot Operating System).

6.3 Robot Operating System

Robot operating system (ROS) is a set of general-purpose software library, which can be used with PX4 autopilot software by using offboard mod. The MAVROS MAVLink node is used to communicate with PX4 using the Gazebo simulator or running on hardware. MAVROS acts as the bridge between the PX4 flight stack and ROS enabled companion computer. MAVROS sends the messages provided by the PX4 autopilot as the ROS topic in the MAVLink message format. The key functions of MAVROS as shown in Fig. 19 is the ability to send commands such as attitude, position, and velocities topics namely to the UAV through an offboard mode which is fully supported in the PX4.

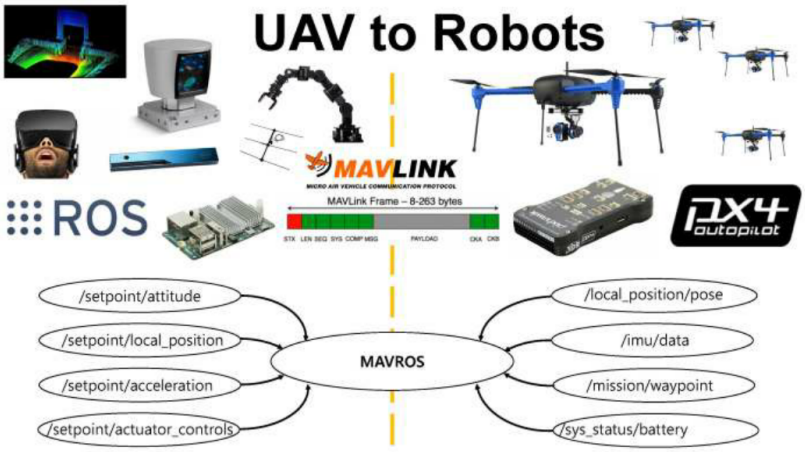

Figure 19: Robot Operating System with MAVROS [32].

6.4 Robot Operating System with Gazebo Simulation

Robot operating system (ROS) fully can be compatible with PX4 and the Gazebo simulator. The pattern of the ROS integration with PX4 and Gazebo simulator is shown in Fig. 20. It shows PX4-based simulation environment. PX4 receives some sensor data from the Gazebo simulator which is a physics engine, PX4 sends motor and actuator command to the simulator. In addition, it can communicate with the ground control station (GCS) such as QGroundControl, which indicates the condition of the vehicle.

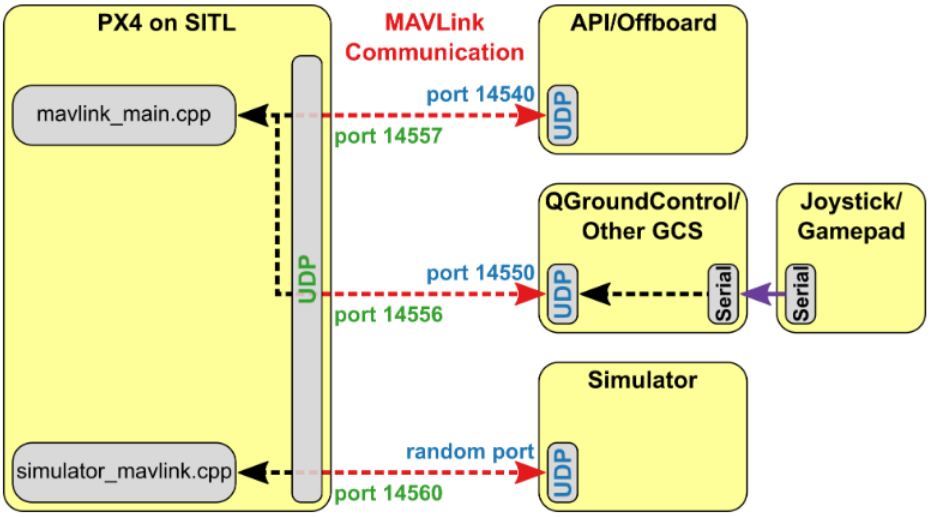

Figure 20: ROS/Gazebo integration with PX4 [31].

6.5 Robotics System Toolbox In Simulink

MATLAB Simulink environment was used for developing and testing the algorithm. MATLAB Simulink environment supports a robotics system toolbox. The robotics system toolbox is compatible with ROS. It is possible for MATLAB Simulink to communicate with Linux embedded computer and a commercially available autopilot by using ROS. The robotics system toolbox takes less time to develop and test algorithms than previous methods.

Figure 21 shows the MATLAB Simulink diagram used in this paper. The robotics system toolbox includes five blocks such as publish, subscribe, blank message, get parameter and set parameter. Both topics and parameters can be published and subscribed through the robotics system toolbox which can use ROS. Publish block is to publish to ROS the topic such as the attitude as the command. The ROS topic such as attitude which is used in this study as shown in Fig. 21 keeps the attitude of the vehicle. Subscribe block is to subscribe to the ROS topic. ROS topics can be selected from a list of topics in the ROS network and the new ROS topic can be defined by the user.

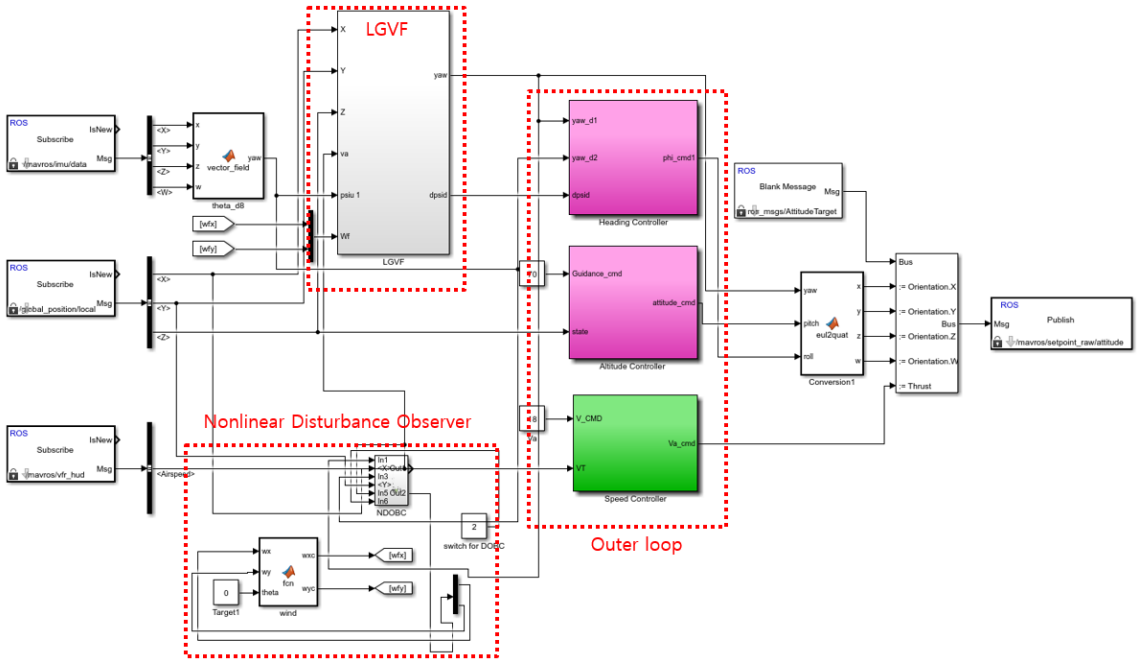

Figure 21: MATLAB Simulink diagram.

Figure 22 shows the outer loop controller structure with anti-windup in the MATLAB Simulink environment. The anti-windup works to relieve the accumulation of integrators. The outer loop's heading, speed, and altitude controllers have the same structure. The outer loop controller calculates the error between the guidance command and the output state such as heading, speed, and altitude. It uses a proportional-integral (PI) controller with anti-windup in order to generate the desired attitude command such as roll, pitch, and throttle which is sent to PIXHAWK running PX4 autopilot in the form of attitude topic using the MAVROS MAVlink node.

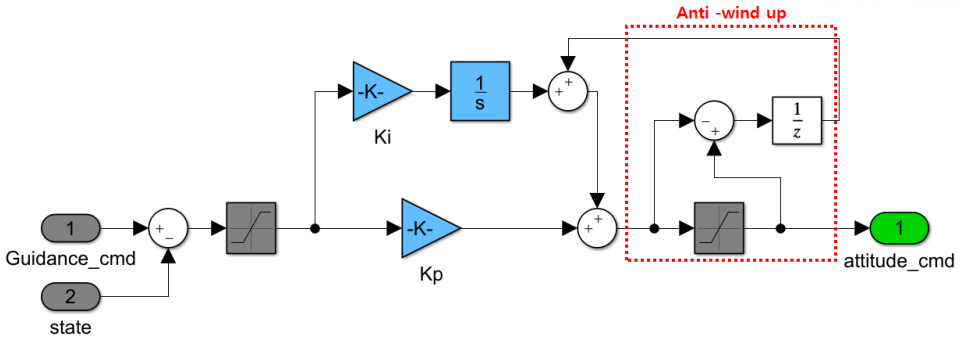

Figure 22: Outer loop controller with anti-wind up.

6.6 Software In The Loop Configuration

The SITL configuration used in this study is shown in Fig. 23. The simulation configuration includes the proposed algorithm, the PX4 autopilot software, and the Gazebo simulator. The proposed algorithm is tested in the MATLAB Simulink environment which has compatibility with the robotics system toolbox. The PX4 autopilot software is used to stabilize the inner loop controller. The MATLAB Simulink and PX4 autopilot can communicate with each other by using MAVROS MAVLink. The Gazebo simulator simulates using a small fixed-wing UAV model.

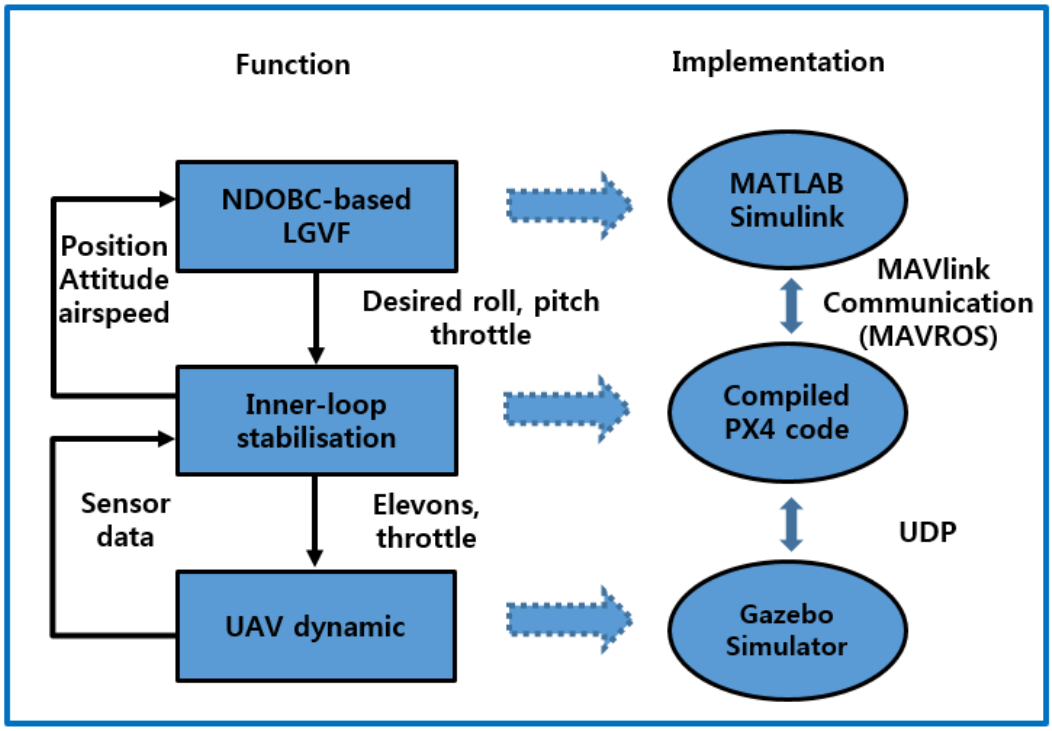

Figure 23: Software in the loop configuration.

The MATLAB Simulink environment implements the NDOBC-based LGVF path following controller. The outer loop controllers such as heading, altitude and speed controller are tuned during the simulation test. The desired attitude such as roll and pitch from the outer loop controller is sent to the PX4 autopilot. The PX4 autopilot is an open-source flight control software for the UAV. In this study, The PX4 autopilot is used to follow the attitude command such as the desired roll, pitch from the NDOBC-based LGVF controller. PX4 autopilot and Gazebo simulator are connected through the UDP network connection. MATLAB Simulink environment can communicate with the PX4-based Gazebo simulator by using the ROS. The flight data are transferred through ROS topics using MAVROS.

Table 2: Gain of Outer Loop Controller.

Gain	K_i	K_p
Heading Controller	0.017	0.85
Altitude Controller	0.003	0.03
Speed Controller	0.25	0.65

6.7 Simulation Result

Prior to SITL simulation using NDOBC and LGVF methods, The SITL simulation is first performed to confirm the performance of the outer loop controller that controls the guidance command such as the heading, altitude, and speed. The outer loop controller with proportional integral (PI) and anti-wind up is tuned during the test.

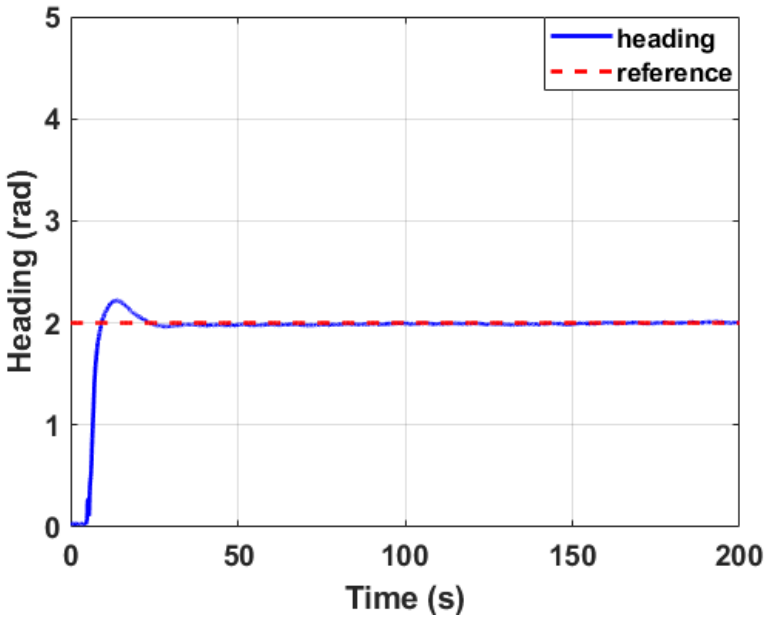

Figure 24: Heading.

The heading, altitude, and speed command of the UAV are 2rad, 50m, and 22m/s in Figs. 24, 25, and 26, respectively. It is shown that the UAV follows the command well in Figs. 24, 25, and 26, respectively.

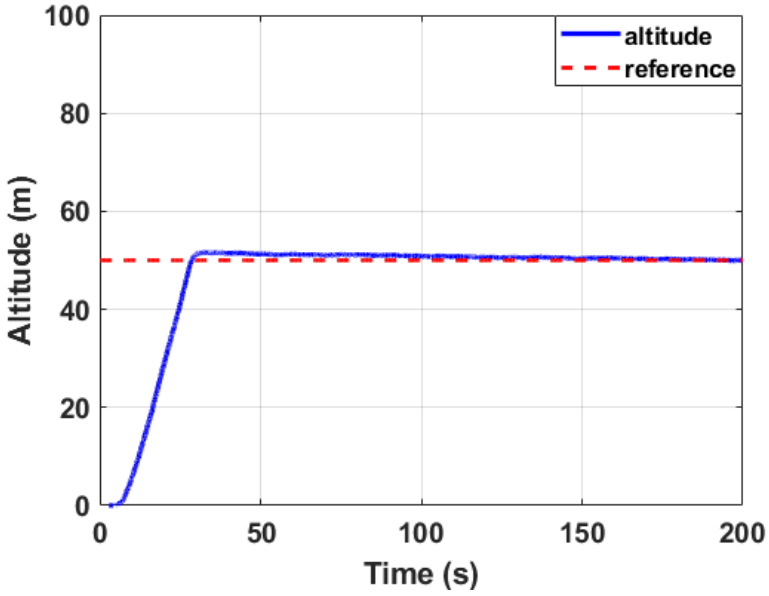

Figure 25: Altitude.

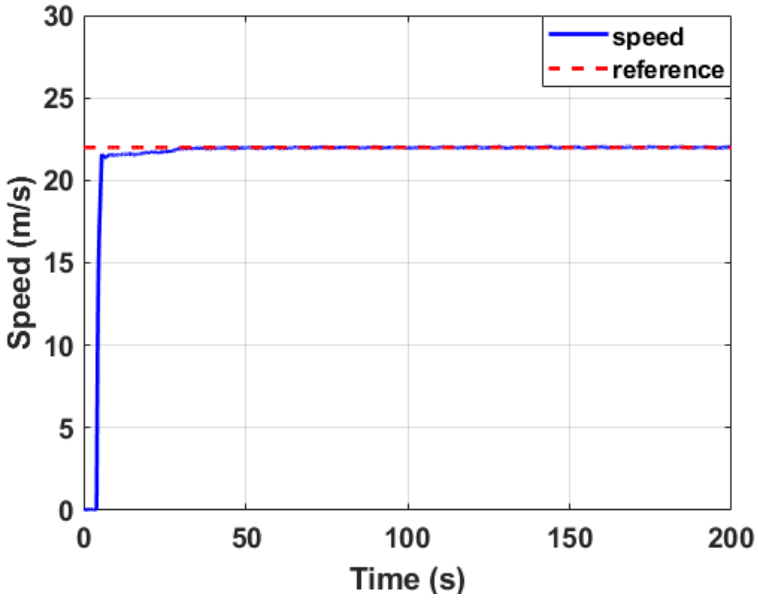

Figure 26: Speed.

After confirming the performance of the heading, altitude, and speed controller of the outer loop in Figs. 24, 25, and 26, SITL simulation using NDOBC and LGVF methods are carried out in the environment where there is no disturbance such as wind. In Fig. 27, the disturbance estimated by the nonlinear disturbance observer is almost zero. It is shown that the circular pattern following performance is almost the same as that of the LGVF without a nonlinear disturbance observer. The advantage of DOBC is that the nominal performance of the designed controller is recovered in the absence of disturbances or uncertainties.

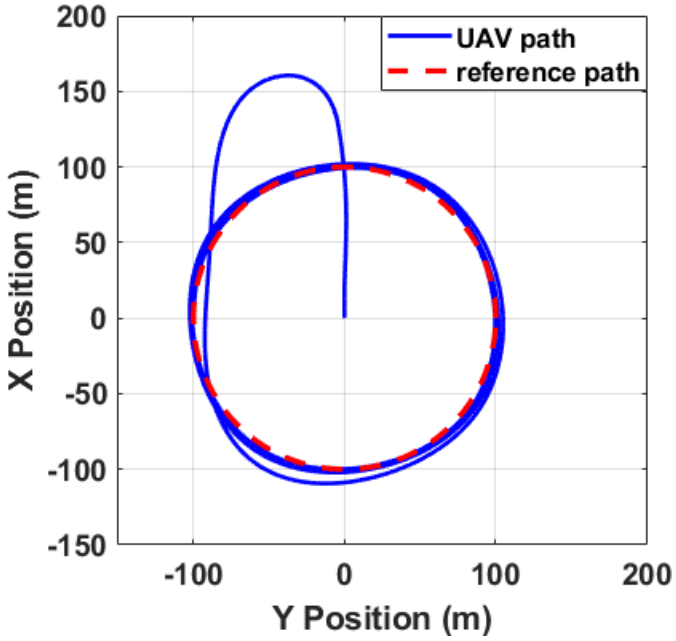

Figure 27: UAV path.

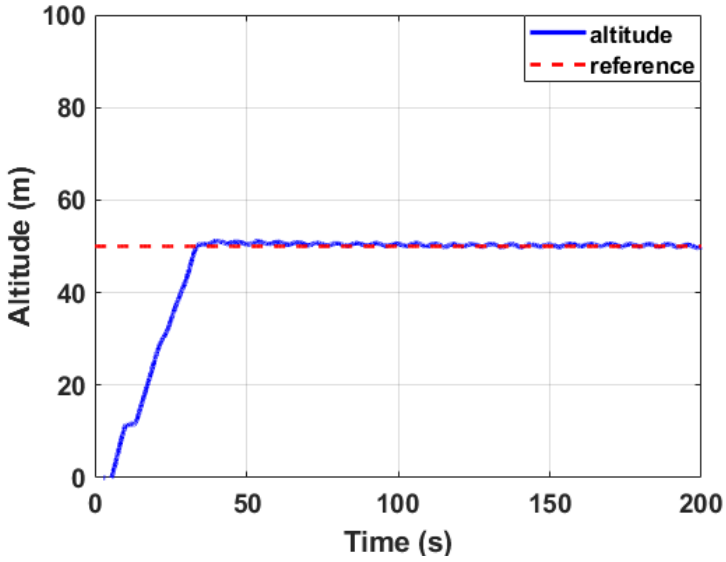

Figure 28: Altitude.

For safe takeoff of the UAV, the algorithm in MATLAB Simulink environment is implemented

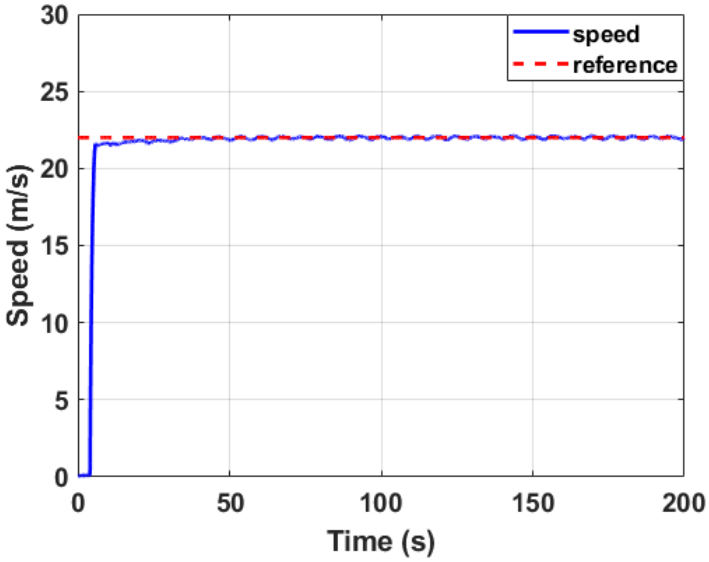

Figure 29: Speed.

so that the UAV's heading command from outer loop controller operates after the altitude reaches 10m. It is shown that the UAV follows the command of altitude and speed well in Figs. 28 and 29, respectively.

VII Outdoor Flight Experiment

Outdoor flight experiments are performed out based on the software in the loop simulation algorithm which can be deployed in the small fixed-wing UAV in wind condition at approximately 3m/s.

7.1 Skywalker X-5 UAV

Figure 30 shows a Skywalker X-5 which is the small fixed-wing UAV used in this study. In flight experiment configuration, the weight and wingspan of the Skywalker X-5 are 1.5kg and 1.18m. The X-5 UAV is equipped with PX4-based PIXHAWK autopilot hardware and an on-board computer such as Raspberry Pi.

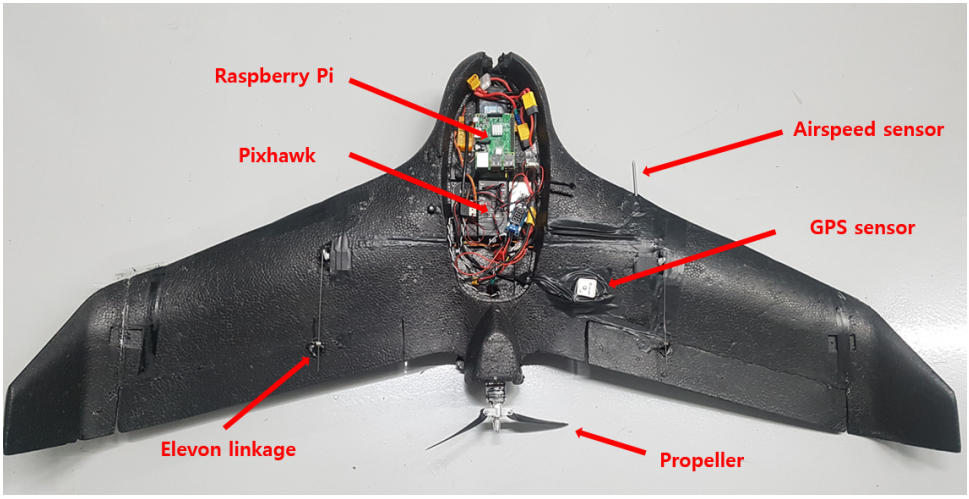

Figure 30: The Skywalker X-5 used in the outdoor flight experiment.

7.2 Auto Code Generation

Auto code generation from the model as shown in Fig. 31 is one of the key features of MATLAB Simulink. Auto code generation is to automatically transfer MATLAB Simulink models into

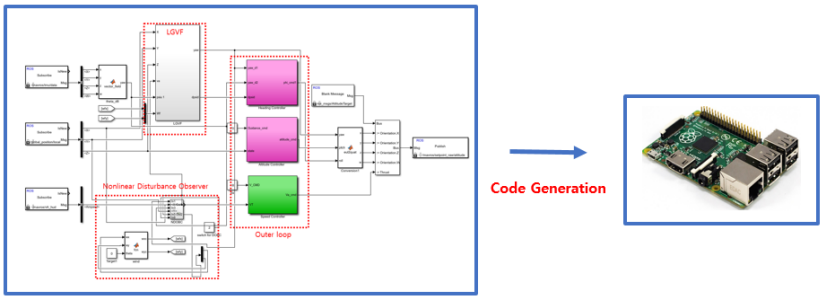

Figure 31: Code generation.

C/C++ code so that it can be deployed on on-board computer such as Raspberry Pi which do not support Simulink models. Powerful feature of auto code generation is that communication delays are greatly reduced by putting code on-board computer of the UAV. The proposed algorithm of MATLAB Simulink environment is deployed in Raspberry Pi device which is embedded systems through auto code generation.

7.3 Flight Experiment Configuration

Figure 32 shows a flight experiment configuration used in this study. The LGVF path following and the NDOBC algorithm are executed in Raspberry Pi. The desired attitude commands such as roll, pitch, and throttle computed from the path following controller are sent to PX4-based PIXHAWK which executes PX4 autopilot through MAVROS MAVLink. PX4-based PIXHAWK tracks the attitude commands to stabilize the inner loop and sends the sensor data such as position, altitude, and airspeed to Raspberry Pi.

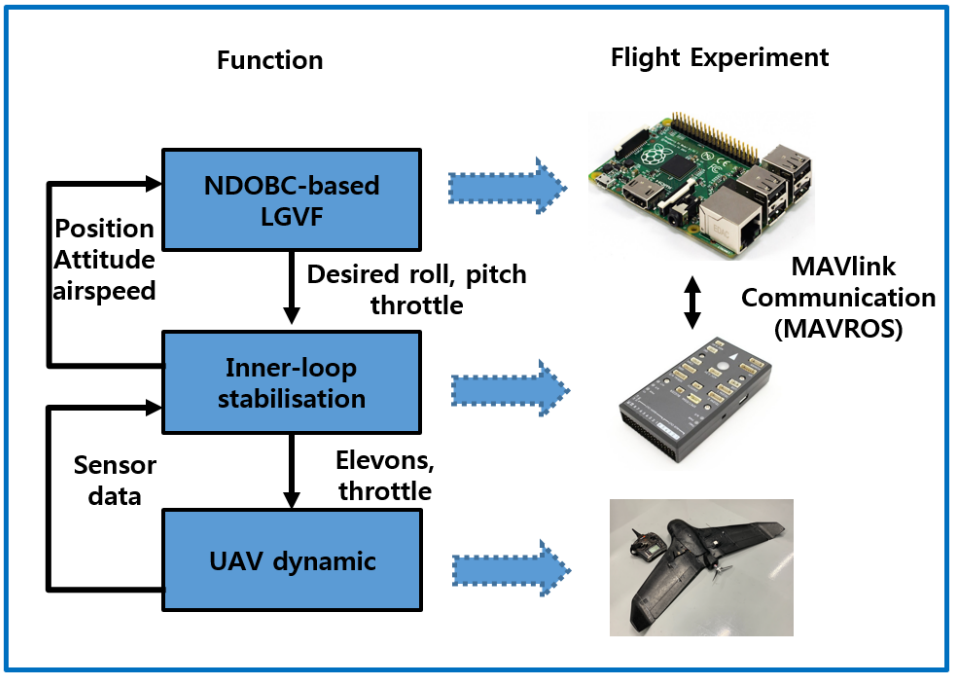

Figure 32: Flight experiment configuration.

Table 3: Gain of Outer Loop Controller.

Gain	K_i	K_p
Heading Controller	0.017	0.5
Altitude Controller	0.004	0.04
Speed Controller	0.025	0.35

7.4 Experiment Result

Figure 33 shows the scenery of the experimental site. The skywalker X-5 UAV takes off in the manual mode of the RC controller until the altitude is close to 50m, then it is switched as the automatic mode to perform the proposed algorithm.

Figure 33: Flight experiment of skywalker X-5 UAV.

Figure 34 and 35 show the flight experiment results for the X-5 UAV using LGVF without and with DOBC. The weather condition was very similar for both cases at approximately 3m/s. In Fig. 34, it is shown that the X-5 UAV using LGVF without DOBC follows the path far away from the reference path due to the effect of wind. In Fig. 35, the X-5 UAV using LGVF with DOBC follows the reference path better. The DOBC helps to reject wind disturbances and result in the accurate path following performance.

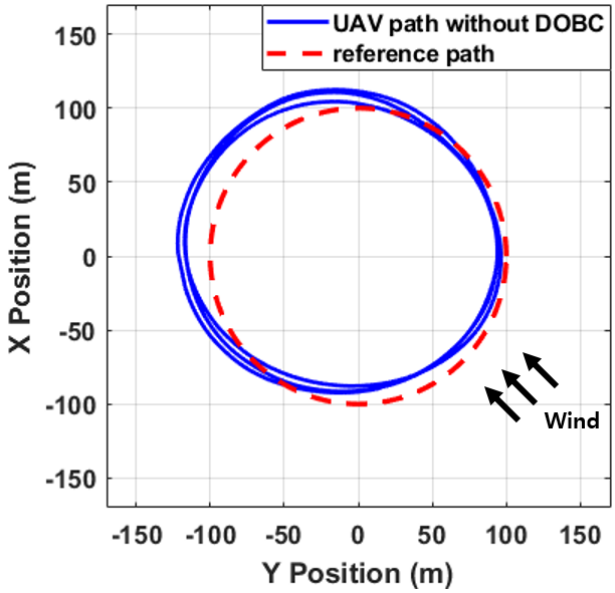

Figure 34: Flight experiment without DOBC.

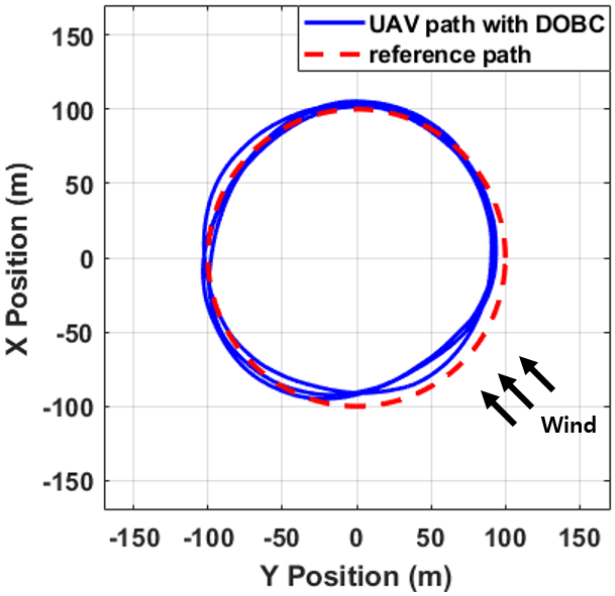

Figure 35: Flight experiment with DOBC.

Figure 36 shows that the airspeed and ground speed measured from the sensors of PIXHAWK which includes airspeed and GPS sensors. Figure 37 shows the disturbance estimated from the nonlinear disturbance observer, and disturbance such as wind which is calculated using the difference between airspeed and ground speed measured from PIXHAWK sensors. The estimated disturbance and disturbance show almost the same tendency, and the average values of the estimated disturbance and disturbance are 3.4m/s and 2.6m/s, respectively. Here, the estimated disturbance includes the modeling error by the system uncertainty, and sensor noise as well as disturbance such as wind.

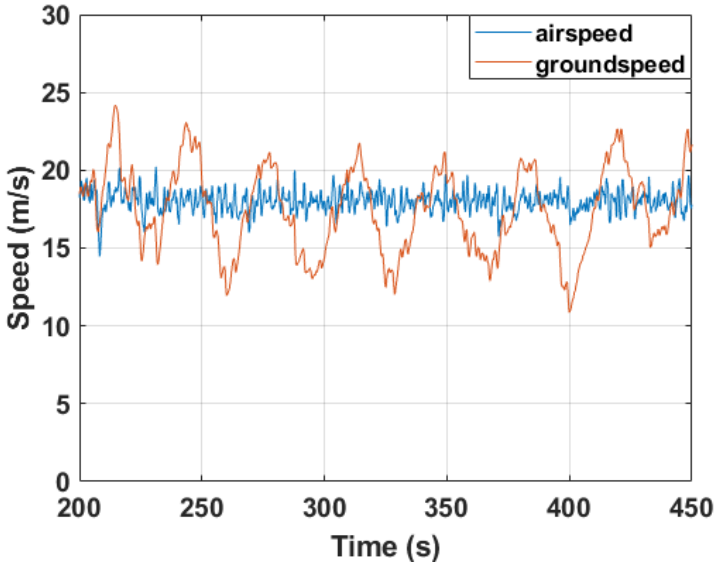

Figure 36: Airspeed and groundspeed.

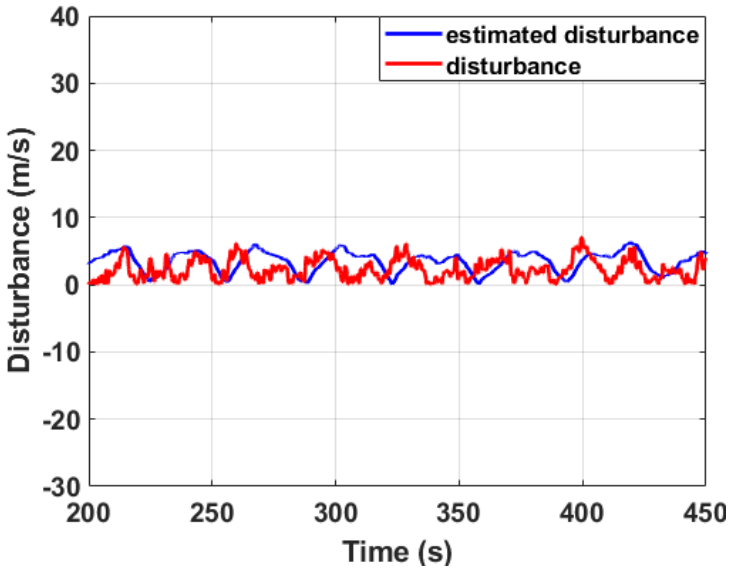

Figure 37: Disturbance.

The speed and altitude commands of the X-5 UAV are 18m/s and 70m, respectively. The outer loop controller gains such as speed controller and altitude controller gains are tuned during the outdoor flight experiment. It is shown that the X-5 UAV follows the command well in Figs. 38 and 39, respectively.

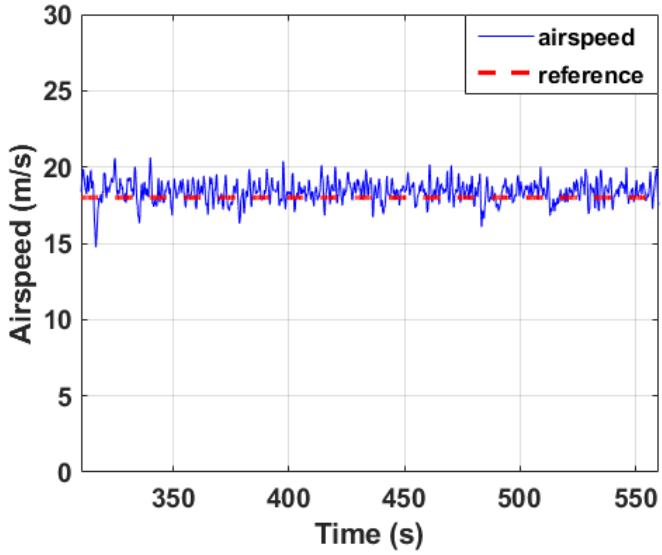

Figure 38: Airspeed.

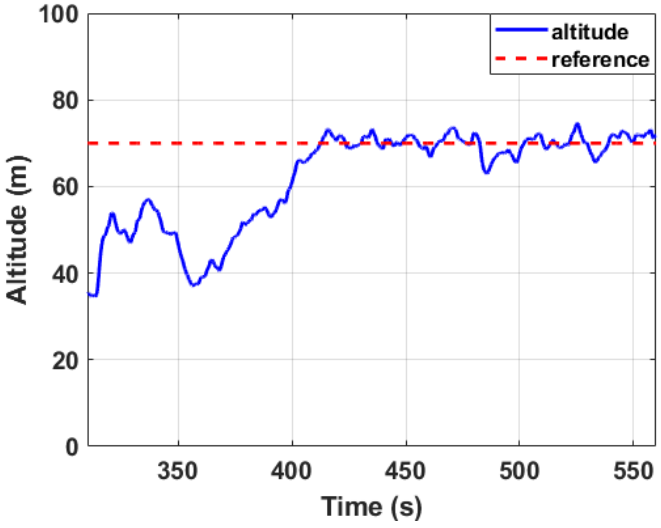

Figure 39: Altitude.

Figure 40 and 41 show that the X-5 UAV follows the attitude commands such as pitch and roll. The PX4-based PIXHAWK autopilot stabilizes the inner loop.

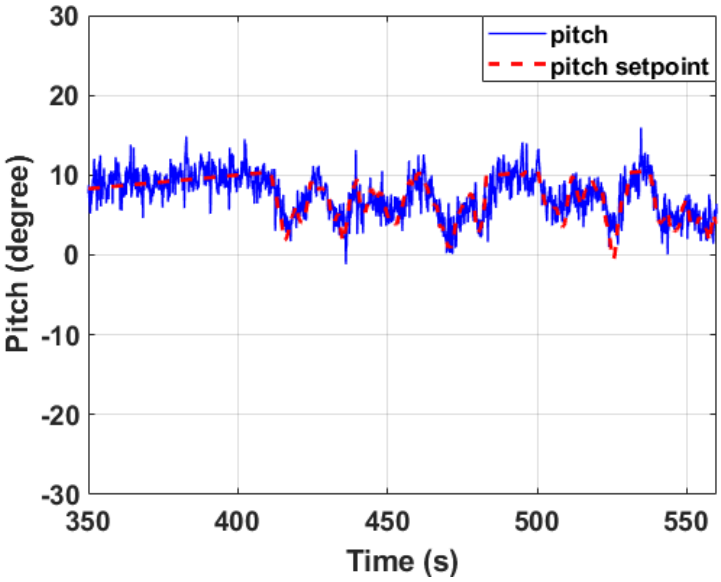

Figure 40: Pitch.

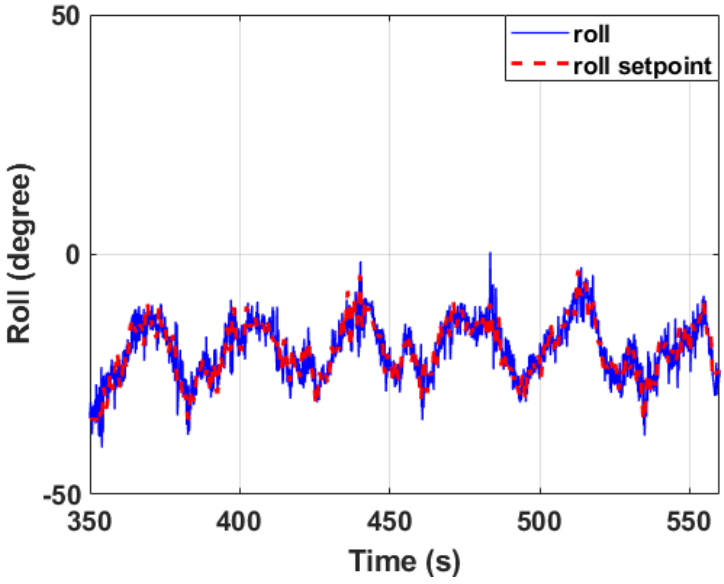

Figure 41: Roll.

VIII Conclusion and Future work

This paper proposes a nonlinear disturbance observer-based LGVF path following controller for a small fixed-wing UAV affected by disturbance such as wind. The proposed controller incorporates wind estimates into the LGVF path following controller. NDOBC method for a small fixed-wing UAV was used to estimate and compensate for the disturbance in order to follow the circular path accurately using the LGVF technique.

Numerical simulation through MATLAB Simulink environment is first carried out to verify its performance. The software in the loop simulation is carried out to verify the performance of the proposed algorithm which can be deployed in the small fixed-wing UAV. Real flight experiments using the X-5 UAV demonstrated the performance of the proposed path following method.

Future work is to apply the DOBC method to the bird-inspired UAV which is currently researching in our laboratory. The bird-inspired UAV is susceptible to disturbance such as wind. The LGVF method is mainly used to standoff tracking, and we plan to research the standoff tracking by applying the DOBC method to the bird-inspired UAV.

References

[1] Park, S., Deyst, J., and How, J. P. (2007). Performance and Lyapunov stability of a nonlinear path following guidance method. Journal of Guidance, Control, and Dynamics, 30(6), 1718-1728

[2] Kothari, M., Postlethwaite, I., and Gu, D. W. (2010). A Suboptimal Path Planning Algorithm Using Rapidly-exploring Random Trees. International Journal of Aerospace Innovations, 2.

[3] Ratnoo, A., Sujit, P. B., and Kothari, M. (2011). Adaptive optimal path following for high wind flights. IFAC Proceedings Volumes, 44(1), 12985-12990.

[4] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W. (2007). Vector field path following for miniature air vehicles. IEEE Transactions on Robotics, 23(3), 519-529.

[5] Sujit, P. B., Saripalli, S., and Sousa, J. B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. IEEE Control Systems Magazine, 34(1), 42-59.

[6] Watkins, S., Milbank, J., Loxton, B. J., and Melbourne, W. H. (2006). Atmospheric winds and their implications for microair vehicles. AIAA journal, 44(11), 2591-2600.

[7] Frew, E. W., Lawrence, D. A., and Morris, S. (2008). Coordinated standoff tracking of moving targets using Lyapunov guidance vector fields. Journal of guidance, control, and dynamics, 31(2), 290-306.

[8] Oh, H., Kim, S., Tsourdos, A., and White, B. A. (2014). Decentralised standoff tracking of moving targets using adaptive sliding mode control for UAVs. Journal of Intelligent and Robotic Systems, 76(1), 169-183.

[9] Oh, H., Kim, S., Shin, H. S., and Tsourdos, A. (2015). Coordinated standoff tracking of moving target groups using multiple UAVs. IEEE Transactions on Aerospace and Electronic Systems, 51(2), 1501-1514.

[10] Kaminer, I., Yakimenko, O., Pascoal, A., and Ghabcheloo, R. (2006, June). Path generation, path following and coordinated control for timecritical missions of multiple UAVs. In 2006 American Control Conference (pp. 4906-4913). IEEE.

[11] Liu, C., Chen, W. H., and Andrews, J. (2012). Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers. Control Engineering Practice, 20(3), 258-268.

[12] Lee, S. J., Kim, S., Johansson, K. H., and Kim, H. J. (2016, December). Robust acceleration control of a hexarotor UAV with a disturbance observer. In 2016 IEEE 55th Conference on Decision and Control (CDC) (pp. 4166-4171). IEEE.

[13] Besnard, L., Shtessel, Y. B., and Landrum, B. (2012). Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. Journal of the Franklin Institute, 349(2), 658-684.

[14] Lee, K., Back, J., and Choy, I. (2014). Nonlinear disturbance observer based robust attitude tracking controller for quadrotor UAVs. International Journal of Control, Automation and Systems, 12(6), 1266-1275.

[15] Wang, H., and Chen, M. (2016). Trajectory tracking control for an indoor quadrotor UAV based on the disturbance observer. Transactions of the Institute of Measurement and Control, 38(6), 675-692.

[16] Liu, C., and Chen, W. H. (2016). Disturbance rejection flight control for small fixed-wing unmanned aerial vehicles. Journal of Guidance, Control, and Dynamics, 2810-2819.

[17] Smith, J., Su, J., Liu, C., and Chen, W. H. (2017). Disturbance observer based control with anti-windup applied to a small fixed wing UAV for disturbance rejection. Journal of Intelligent and Robotic Systems, 88(2-4), 329-346.

[18] Liu, C., McAree, O., and Chen, W. H. (2013). Path-following control for small fixed-wing unmanned aerial vehicles under wind disturbances. International Journal of Robust and Nonlinear Control, 23(15), 1682-1698.

[19] Yang, J., Chen, W. H., and Li, S. (2010). Autopilot design of bank-to-turn missiles using state-space disturbance observers.

[20] Chen, W. H., Ballance, D. J., Gawthrop, P. J., and O'Reilly, J. (2000). A nonlinear disturbance observer for robotic manipulators. IEEE Transactions on industrial Electronics, 47(4), 932-938.

[21] Chen, W. H. (2003). Nonlinear disturbance observer-enhanced dynamic inversion control of missiles. Journal of Guidance, Control, and Dynamics, 26(1), 161-166.

[22] Chen, W. H. (2004). Disturbance observer based control for nonlinear systems. IEEE/ASME transactions on mechatronics, 9(4), 706-710. ‘

[23] Chen, W. H., Ballance, D. J., Gawthrop, P. J., Gribble, J. J., and O'Reilly, J. (1999). Nonlinear PID predictive controller. IEE Proceedings-Control Theory and Applications, 146(6), 603-611.

[24] Lawrence, D. (2003, September). Lyapunov vector fields for UAV flock coordination. In 2nd AIAA" Unmanned Unlimited" Conf. and Workshop and Exhibit (p. 6575).

[25] Frew, E. W., Lawrence, D. A., Dixon, C., Elston, J., and Pisano, W. J. (2007, July). Lyapunov guidance vector fields for unmanned aircraft applications. In 2007 American control conference (pp. 371-376). IEEE.

[26] Kingston, D., and Beard, R. (2007). UAV splay state configuration for moving targets in wind. In Advances in Cooperative Control and Optimization (pp. 109-128). Springer, Berlin, Heidelberg.

[27] Oh, H., Kim, S., and Tsourdos, A. (2010). Coordinated Standoff Tracking of Moving Ground Targets Using Multiple UAVs. Encyclopedia of Aerospace Engineering, 1-14.

[28] Li, S., Yang, J., Chen, W. H., and Chen, X. (2016). Disturbance observer-based control: methods and applications. CRC press.

[29] Shim, H., Park, G., Joo, Y., Back, J., and Jo, N. H. (2016). Yet another tutorial of disturbance observer: robust stabilization and recovery of nominal performance. Control Theory and Technology, 14(3), 237-249.

[30] Shim, H., and Joo, Y. J. (2007, December). State space analysis of disturbance observer and a robust stability condition. In 2007 46th IEEE Conference on Decision and Control (pp. 2193-2198). IEEE.

[31] PX4 v1.8.2 Developer Guide
`https://dev.px4.io/en`

[32] Robot Operating Sytem(ROS)
`https://www.ros.org/`

Acknowledgements

I submitted my thesis on the research topic I have been studying during my master's course. There are a lot of people who have helped me over the past two years. It was so fortunate to have met these people in my life, and I would like to thank them.

I would like to first thank my advisor, Dr. Hyondong Oh, who has always provided me help and guidance throughout my studies. I am very thankful for the attention and advice that he has been able to give and guide me through the right way. I would also like to thank the people who worked together in the autonomous systems laboratory (ASL). Thanks to the people, I could learn a lot and have fun during my studies.

Finally, I would like to thank my family for acting as a supporter for me during my master's course. Thank you very much for your support.