

기계학습 알고리즘을 이용한 UAS 제어계수 실시간 자동 조정 시스템

UAS Automatic Control Parameter Tuning System using Machine Learning Module

문미선*, 송 강*, 송동호*

Mi-Sun Moon*, Kang Song* and Dong-Ho Song*

요 약

무인기의 자동 비행 제어 시스템은 기체의 형태, 크기, 무게 등의 정적 및 동적 변화에 따라 스스로 비행계수를 조정하여 목표 비행궤적을 정확히 따라가도록 제어할 필요가 있다. 본 논문에서는 PID 제어 기법을 이용하는 비행제어시스템에 기계학습모듈(MLM)을 추가하여 기체의 특성 변화에 따라 제어계수를 비행중 실시간 자동적으로 조정하는 시스템을 제안한다. MLM은 선형회귀분석과 보정학습을 이용하여 설계되었으며 MLM을 통해 학습된 제어계수의 적합성을 평가하는 평가모듈(EvM)을 함께 모델링 하였다. 이 시스템은 FDC 비버 시뮬레이터를 기반으로 실험하였으며 그 결과를 분석 제시하였다.

Abstract

A automatic flight control system(AFCS) of UAS needs to control its flight path along target path exactly as adjusts flight coefficient itself depending on static or dynamic changes of airplane's features such as type, size or weight. In this paper, we propose system which tunes control gain autonomously depending on change of airplane's feature in flight as adding MLM(Machine Learning Module) on AFCS. MLM is designed with Linear Regression algorithm and Reinforcement Learning and it includes EvM(Evaluation Module) which evaluates learned control gain from MLM and verified system. This system is tested on beaver FDC simulator and we present its analysed result.

Key words : automatic control gain tuning, machine learning, numeric prediction

I. 서 론

무인항공기(UAS, Unmanned Aircraft System)의 자동 비행 제어 시스템은 기체의 형태, 크기, 무게 등의 정적 및 동적 변화에 따라 스스로 적응을 할 필요가

있다. 이는 한 비행제어컴퓨터를 유사기종의 크기가 다른 기체에 적용할 때와 동일기체의 유료하중(payload)이 주어진 범위를 초과하는 경우에 흔히 발생할 수 있다. 유사기종의 크기가 다른 기체는 기체의 크기에 따라 비행 특성의 차이로 인하여 액추에이

* 한국항공대학교 컴퓨터공학과

· 제1저자 (First Author) : 문미선

· 투고일자 : 2010년 10월 11일

· 심사(수정)일자 : 2010년 10월 12일 (수정일자 : 2010년 12월 13일)

· 게재일자 : 2010년 12월 30일

* 2009년 한국항공학회 학술대회 발표 논문이며, 논문 심사를 통과하여 게재하는 논문임.

터의 조작량이 같더라도 기체의 움직임 반응이 서로 다르고, 그에 따른 비행 제어 계수가 달라진다. 또한 탑재 화물의 무게가 탑재화물의 중량범위를 넘어서는 경우, 즉, 과도한 화물을 싣거나 이 화물을 비행 중 투하함에 따른 기체충중량의 변화 등, 범위를 벗어난 기체충중량이 비행 동역학의 특성에 커다란 변화를 일으키게 된다. 이런 시스템의 특성이 변하는 경우에 대응하기 위해서는 소프트웨어 교체나 제어 관련 수식 수정이 비행 중에 가능하도록 하는 자가 적응성(Self-Adaptive) 비행 제어 시스템이 필요하다.

본 논문에서는 선형회귀분석(LR, Linear Regression)을 적용하여 하나의 기체(A)에 최적화 되어 있는 제어계수를 특성이 다른 기체(B)에 적용한 후 기체 B의 최적 제어 계수를 자동으로 조정할 수 있는 제어 계수 기계 학습 모듈(MLM, Machine Learning Module)을 설계하고 실험을 통해 시스템의 가능성을 검증한다.

이후 2장에서는 제어 계수 자동 조정에 대한 관련 연구들을 살펴보고, 3장에서는 MLM 구성 및 알고리즘에 대해 설명한다. 4장에서는 설계된 모델에 대한 실험을 하고 그 결과를 분석하고, 5장에서 결론을 맺는다.

II. 관련연구

2-1 PID 제어 계수 자동 조정에 관한 연구

비례적분미분(PID, Proportional-Integral-Derivative) 제어 이론[1]은 매우 오래된 제어 기법이다. 따라서 다양한 분야에서 PID 제어를 사용하고 있고, 최근에는 퍼지 이론과 접목한 제어 이론에 대해서도 연구되고 있다. 또한 각 시스템의 특성에 맞는 PID 모델을 효율적으로 찾기 위해 자동으로 계수를 조정하는 것에 대한 연구는 계속해서 진행되고 있다[2]. 그러나 이러한 계수 조정 방법들은 대부분 시스템이 운용되는 동안 동적으로 계수 조정하는 것이 아니라 시스템 운용 후 결과 데이터를 분석하여 계수를 조정하고, 조정된 계수를 다시 시스템에 적용하여 검증하는 방식으로 이루어진다. 가장 널리 사용되는 알고리즘은

유전자 알고리즘(GA, Genetic Algorithm)로 반복적인 연산을 통해 최적화된 해를 찾는다. 여러 연구에서 제어 매개변수를 자동으로 학습하는 시스템[3]이나 퍼지 PID 제어기의 룰을 줄이고 자동으로 조정하는 시스템[4] 등에 GA를 적용하고 있다. 또한, 신경망 알고리즘(NN, Neural Network algorithm)도 오일 정제 제어 시스템[5]이나 퍼지 논리 제어 시스템 [6], 헬리콥터의 측면 제어[7] 등에 NN을 많이 사용되고 있다. 그러나 이러한 알고리즘들은 많은 연산을 요구하고 있어 소형, 경량, 저전력을 요구하는 UAS 등과 같은 실시간 시스템에는 적합하지 않아 새로운 알고리즘에 대한 모색이 필요하다.

2-2 UAS에서의 계수 자동 조정에 관한 연구

UAS 자동비행제어에서도 PID 제어 이론은 강력한 제어 효율을 보인다[8]. 대부분의 UAS가 PID 제어를 이용하고 있고, 수동으로 계수(gain)을 조정하거나 오프라인으로 데이터를 입력하여 결과를 얻어내는 자동 조정 소프트웨어를 이용한다. 이들 방법은 대부분 수차례의 비행실험을 요구하여 많은 시간과 고비용을 요구한다. 이 고비용 문제를 해결하기 위하여 실시간으로 PID 제어 계수를 GA를 이용하여 자가 조정하는 연구[9]가 시도되고 있으나 부모, 자식을 각 1개씩 취하고 돌연변이(mutation) 연산만 가능한 변형된 GA로 그 정확도가 떨어지며, 다른 연구들 또한 소형 UAS에 적합하지 않아 새로운 알고리즘에 대한 모색이 필요하다. 이에 본 논문에서는 기계학습 알고리즘을 이용한 UAS 비행제어계수 실시간 자동 조정 시스템을 연구한다.

III. 모델링

기계학습 알고리즘을 이용한 UAS 비행제어계수 실시간 자동 조정 시스템은 일반적인 자동비행제어 시스템의 기본 구조를 기반으로 모델링하였다[10].

그림 1은 전체 구조에 대한 그림이다. 기본 구조에 기계학습모듈(MLM, Machine Learning Module)을 추가하여 제어 계수를 실시간 학습할 수 있도록 한다.

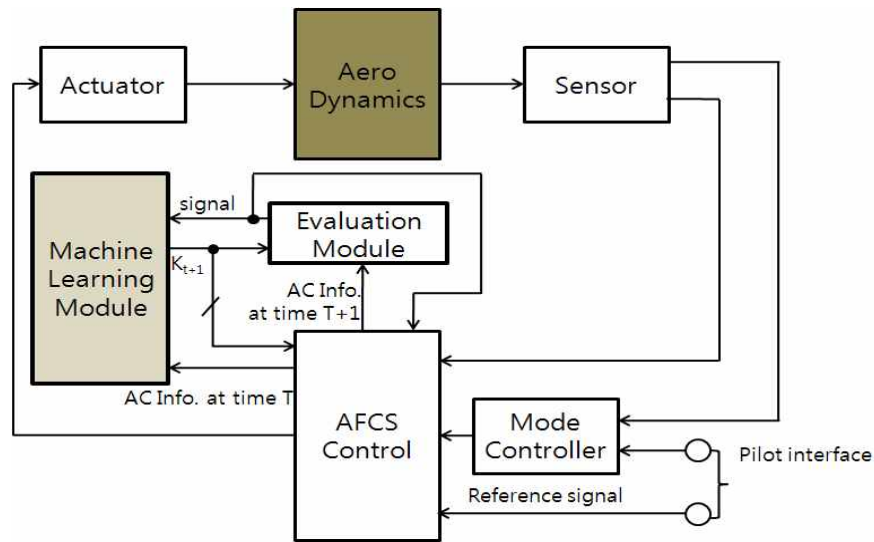


그림 1. 제안 시스템 블록도
Fig. 1. Block Diagram of proposed system

MLM은 비행 제어 시스템으로부터 기체의 자세, 위치, 속도 등의 수치 정보를 전달받아 선형회귀분석(LR, Linear Regression)으로 새로운 제어 계수를 학습하는 모듈이다.

LR은 측정된 비행 데이터를 분석하여 수학적 모델인 선형적 관계식을 구하여 예측하고자 하는 수치, 즉 제어 계수를 구한다.

MLM에서 학습된 새로운 제어 계수는 자동비행 제어시스템(AFCS, Automatic Flight Control System)에 전달되어 비행제어에 적용되고 그 결과를 평가모듈(EvM, Evaluation Module)에서 분석하여 학습된 비행 제어계수를 평가하도록 한다. 이 평가 모듈의 출력으로 MLM의 동작 여부를 결정한다.

3-1 MLM(Machine Learning Module)

MLM의 내부 구조는 그림 2과 같다. 먼저 AFCS가 전달하는 데이터 패킷을 파싱하여 조정할 대상(TUNING type code), 조정에 필요한 데이터(ATTITUDE DATA, POSITION DATA, DESIRED ATTITUDE/POSITION DATA, ACTUATORS Ref.)를 분류하여 이들 데이터를 처리하는 LR 블록 내부의 해당 블록으로 전송한다.

Linear Regression Block

‘Attitude’ 블록은 Roll, Pitch, Yaw의 자세를 제어하는 제어 계수(PID controller's gain)를 조정한다. 기체의 자세 제어는 기체가 평형 상태를 유지하도록 하는 연산을 수행하며 각각의 축을 기준으로 기체의 기울기를 측정하여 그 값이 0이 되도록 기체의 액추에이터를 제어한다.

‘Flight path’ 블록은 자동비행 경로를 제어하는 수식의 계수를 조정한다. 좌우, 앞뒤, 상하 3축의 경로를 제어하는 P, I, D 연산의 계수를 각각 학습하는 블록으로 사용자가 설정한 경로(특정 지점, 수식)를 따라 기체가 비행할 수 있도록 한다.

$$y = K_p \cdot e + K_i \cdot \int e dt \quad (1)$$

식 (1)의 y는 액추에이터를 제어하는 신호이다. 이 값은 식(1)과 같이 PID 식으로부터 연산되며 제어 계수 K_p 와 K_i 에 따라 제어의 정확도 및 속도가 결정된다[2]. 본 논문의 목적은 K_p 나 K_i 와 같은 제어계수를 학습하는 것이므로 식(1)을 K_p 또는 K_i 에 대한 식으로 정리하여 제어 계수를 학습하기 위해 필요한 요소를 추출한다.

$$K_{T+1} = LR(V, y, e, K_T) = \beta_0 + \beta_1 V_T + \beta_2 y_T + \beta_3 e_T + \epsilon_T \quad (2)$$

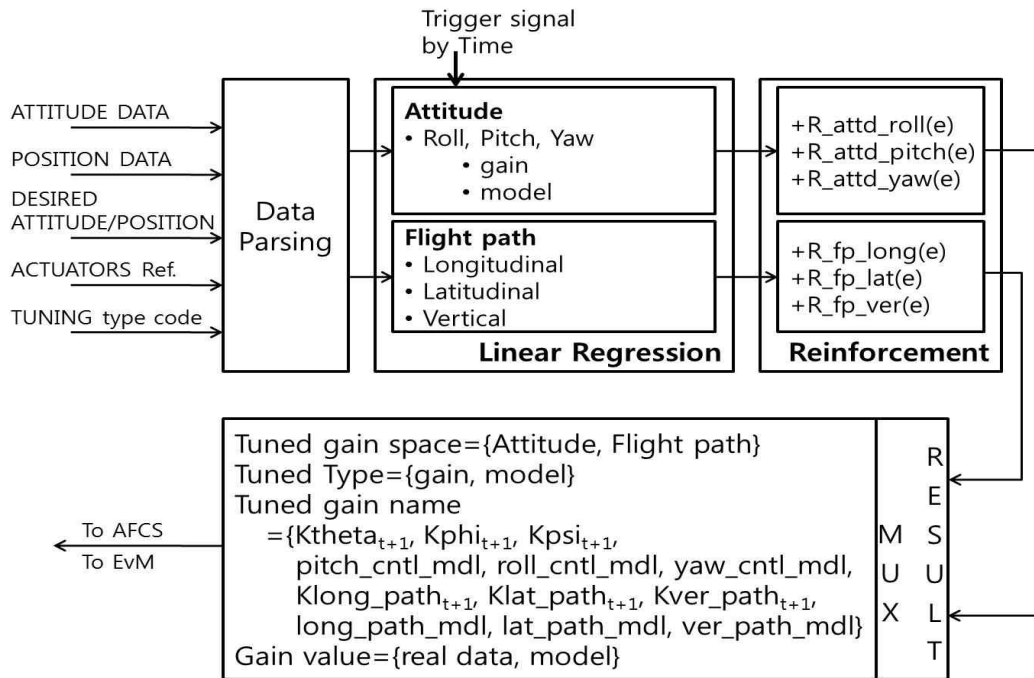


그림 2. MLM 블록도
Fig. 2. MLM block diagram

식(2)는 새로 조정된 제어계수를 구하는 모델이다. K_{T+1} 는 새로 조정된 제어 계수를 의미하고, $LR(V, y, e, K_T)$ 는 학습 알고리즘으로 사용되는 선형 회귀 분석[11, 12] 함수를 나타낸다.

LR과 같은 학습 모듈은 학습 알고리즘의 특성상 데이터의 수집 시간이 필요하므로 일정 시간마다 주기적으로 동작한다. 일정 시간 범위를 $T=[t, t+period)$ 로 하고 이 시간동안 학습할 데이터를 수집한다.

K_{T+1} 은 학습하고자 하는 제어계수이고 V_T, y_T, e_T 는 일정 시간 범위 T 동안 수집된 제어계수 학습에 필요한 요소 데이터이며, 이들 데이터를 분석하여 상수항인 β_0 , 각 요소 데이터의 계수인 $\beta_1, \beta_2, \beta_3$ 를 구한다.

또한 e_T 는 disturbance term으로 임의의 변수다. 이

변수는 잡음(noise)으로 제어계수의 정확성을 저해한다. 따라서 이러한 잡음을 보정하기위해 보정학습 (Reinforcement Learning)[13] 방법을 추가한다.

Reinforcement Learning Block

보정학습은 온라인 수행에 중점을 두고 원하지 않는 행동을 명시적으로 수정하는 학습 방법이다.

그림 3는 보정학습 알고리즘을 나타낸다. 추측한 파라미터가 시스템에 적절한지 확인하고 그 여부에 따라 모델을 수정해나간다.

LR 블록에서 발생한 잡음을 보정학습을 통해 보정하며 식(3)은 식(2)에 보정학습 요소인 보정 함수 $R(e)$ 를 추가한 모델이다. 이 모델은 보정함수를 얼마나 잘 설계하느냐에 따라 학습 시스템의 정확도가 좌우된다.

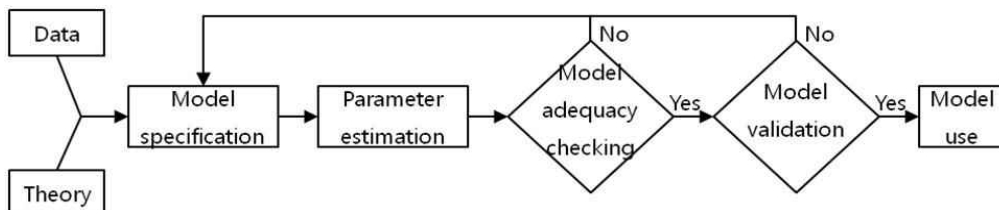


그림 3. Reinforcement learning 알고리즘
Fig. 3. Reinforcement learning algorithm

$$K_{T+1} = R(e) + LR(V, y, e, K_T) \quad (3)$$

MLM은 LR 블록과 Reinforcement 블록을 통해 각 제어 계수를 학습한 후 이 결과를 Merging 블록으로 전송하여 데이터 포맷에 맞추어 AFCS로 보내 시스템 제어에 적용되도록 한다. 또한 EvM(Evaluation Module)로 전송하여 새로이 학습된 제어계수가 시스템에 올바르게 적용되고 있는지 평가한다.

3-2 EvM(Evaluation Module)

EvM은 학습된 새로운 제어 계수가 얼마나 적합한지를 평가하는 모듈이다. 평가 기준은 Error(현재 기체 상태 - desired point)의 크기로 Error의 threshold와 현재 Error를 비교하여 AFCS가 MLM으로부터 계속해서 새로운 제어 계수를 받아들일 것인지 결정한다. desired point는 원하는 비행경로에 대한 수식이나 시간에 따른 특정 지점으로 둘 수 있으며, Error threshold는 기체의 안정성을 유지하는 범위 내로 시스템 설계 시점에서 결정한다.

```

Algorithm : EvM(Evaluation Module)
input : AC state Info., new tuned control gain
output : signal if MLM runs or not.

begin
  initialize error threshold
  receive tuned control gain at time T

  calculate error from AC state Info. packet

  if error is greater than error threshold
    signal is 1
  else
    signal is 0
  end if
end
    
```

이 알고리즘은 Error의 threshold를 얼마로 하느냐에 따라 정확도가 판단된다.

Decision of Error Threshold

학습해야하는 제어계수를 통해 얻어지는 액추에이터의 출력을 기준으로 감내할 수 있는 오차범위를 측정하여 error threshold로 정한다.

자세제어부(inner loop)는 자세 감지 센서 기준으로 ±5°로 한다. 이는 실험을 통해 이 이상의 오차가 발생할 경우 비행 실험이 되지 않음을 확인하였다.

비행경로제어부(outer loop)는 GPS 오차 기준으로 사방 10m로 한다. GPS 오차보다 더 큰 오차가 발생할 경우 지정한 경로를 이탈하여 안정적이 비행이 되지 않으므로 error threshold를 위와 같이 정한다.

IV. 실험

실제 비행체에 실험하는 것은 환경, 구현, 위험성 등의 문제가 있다. 따라서 Beaver 기체를 모델링해 자동비행 환경을 제공하는 FDC 1.3을 시뮬레이터로 이용한다. FDC는 본 연구의 목표인 서로 다른 기체에 적용 가능한 시스템을 구현하기 위해서 서로 다른 기체가 필요하므로 FDC에서 제공하는 Beaver의 특성을 달리하여 다양한 기체를 구현한다. 초기 기체의 상태는 표1에 기술한 속성값을 초기화한다.

표 1. 상태 속성 초기화 리스트
Table 1. The list of initializing state for airplane

속성	값(value)	사용자변경 가능 여부
무게	2288.2Kg	고정
속도	45m/s	시스템제어
고도	500m	시스템제어
heading	0°	시스템제어
RPM	2500	고정
압력/flight-path angle	0	시스템제어
flap angle	0	시스템제어

표1의 속성 중 무게와 RPM은 초기화 시점에서 설정된 값이 변경되지 않는 속성이다. 따라서 이 속성을 다양하게 하여 서로 다른 특성을 가진 기체로 구현할 수 있다.

표 2. 비행모드

Table 2. The flight mode

제어공간	mode	설명
고도	PAH	상승각 유지 비행모드
	ALH	고도 유지 비행모드
	ALS	상승 비행 조정 비행모드
	GS	glideslope
평면	RAH	롤각 유지 비행모드
	HH	방향 유지 비행모드
	LOC	Approach mode
	NAV	자동 제어 모드

표 3. 그룹별 오차 값

Table 3. Error value of each rroup

Group	Error	Group	Error
1	0.019	5	-0.016
2	0.013	6	0.025
3	0.008	7	0.007
4	0.018	8	0.022

선택한 비행 모드는 방향(heading)을 고정시킨 상태에서 고도를 유지하도록 하는 비행모드인 PAH/HH 모드이다.

$$K = -0.001375 * V^2 + 0.1575 * V - 4.8031 \quad (5)$$

기존의 제어 계수는 식(5)와 같은 형태로 속도에 대해서 정의된 수식이다. 위의 수식을 학습 알고리즘에 사용하는 속도(V), 오차(error), 액추에이터 참조값에 대한 수식으로 변경한 후 비행 시뮬레이션을 한다. 시뮬레이션 시간은 5000초이며 이 시간동안 수집된 데이터를 기준으로 학습 알고리즘을 동작시킨다.

$$K = a * V + b * error + c * y + d \quad (6)$$

식(6)은 제어 계수에 대한 새로운 모델이 되며, 이 모델 각 항의 계수는 식 (2)의 $\beta_0, \beta_1, \beta_2, \beta_3$ 를 학습한 결과와 같다.

Experiment : over weight

기본 무게 2288.2Kg이며 안정을 유지하는 최대 무게는 2831.15Kg까지이다. 기본 2288.2Kg를 3500Kg으로 변경한 후 이에 맞는 제어 계수를 학습하는 것이 목표이다.

첫 번째 단계는 기체 무게 2288.2Kg으로 시뮬레이션을 한다. 이는 안정화 된 상태의 결과로 이후 특성 변경된 자동비행제어시스템의 제어 계수를 학습하여 얻은 최종 결과와 비교하는 대상이 된다. 초기의 K를 식 (5)로 설정하여 실행한 결과 데이터로 1차 학습하여 식 (6)과 같은 형태의 제어 계수 식을 설정한다. 이 두 계수 식으로 수행한 시뮬레이션 결과는 그림 4와 같이 일치한다.

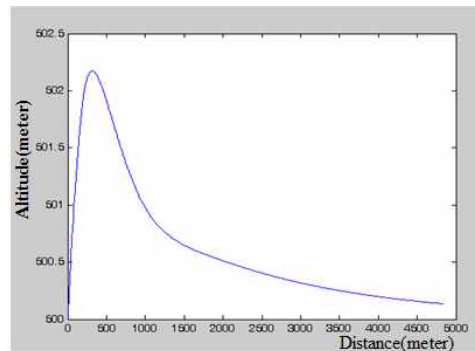


그림 4. 무게 2288.2Kg의 고도 변화(3500Kg의 기준점)
Fig. 4. Altitude of weight 2288.2Kg(desired point of weight 3500Kg)

두 번째 단계는 기체의 무게를 변경한다. 기체의 무게를 3500Kg으로 변경한 후 시뮬레이션을 실행한다. 무거워진 기체는 그림 5와 같이 곧바로 추락하기 때문에 계수 조정을 통해 비행 상태를 안정화 시켜야 한다.

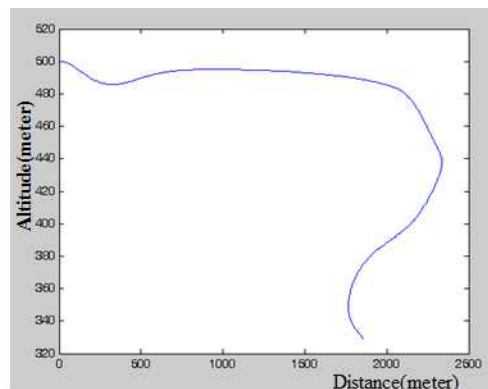


그림 5. 계수 조정 전 무게 3500Kg의 고도
Fig. 5. Altitude of no tuning gain weight 3500Kg

그림 6은 학습 모듈 1회 수행한 결과이다. 1회 학습만으로도 추락에서 벗어나는 결과를 볼 수 있다.

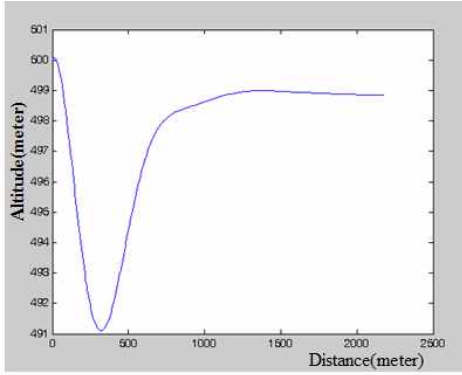
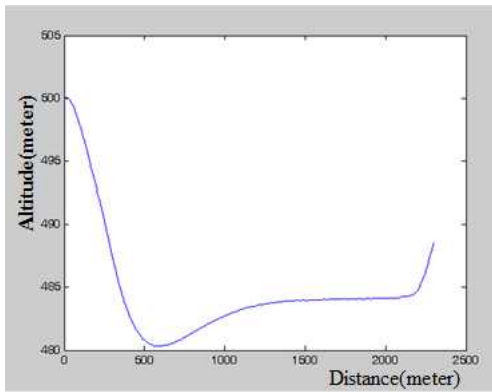
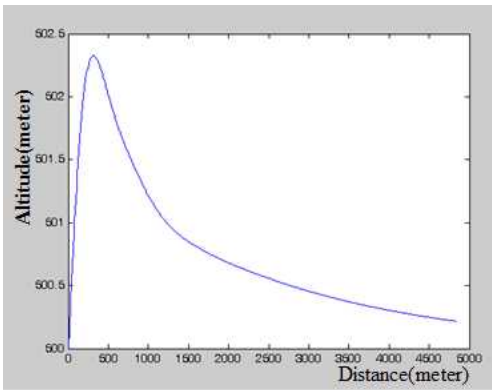


그림 6. 학습 모듈 1회 동작 후 고도 변화
Fig. 6. Altitude of weight 3500Kg after first tuning



(a) 보정 전



(b) 보정 후

그림 7. 10회 학습 후 보정 전, 후 고도 변화
Fig. 7. Altitude of weight 3500Kg after 10 times tuning

LR을 이용하여 학습한 결과와 보정 함수 $R(e)$ 의 값을 더하여 새로운 제어 계수 수식을 얻어 다시 적용한다.

본 논문에서 보정함수 $R(e)$ 는 FDC를 통해 획득한 다량의 데이터를 8 그룹으로 나누어 각각을 LR을 통해 학습한 후 결과를 분석하여 설계한다. 표 3의 각

그룹의 보정값의 평균으로 보정함수 $R(e)$ 는 0.012의 상수 함수로 설계하였다.

그림 7은 학습 모듈을 10회 수행한 후의 결과이다. (a)는 학습 모듈 수행 중 보정함수 $R(e)$ 적용 전 결과이고 (b)는 함수 적용 후 결과이다. 그림 7의 결과로 보정함수 $R(e)$ 의 필요성을 확인할 수 있다.

그림 7에서 보여진 결과는 첫 번째 단계의 결과(그림 4)와 같은 형태인 것을 관찰할 수 있다. 1.2절에서 보인 사전연구에서 실험했던 제어 계수 조정 횟수와 비교해서 줄어든 것을 확인할 수 있다.

V. 결 론

다양한 특성의 여러 UASs 동체에 적용 가능한 AFCs를 설계하고 구현하기 위해서 제어 계수를 자동으로 조정하는 방법을 이용하였다. 기존의 trial-and-error 방법이나 오프라인에서 조정하여 새로운 제어 계수를 적용하여 다시 비행실험을 하도록 하는 방법은 수차례 비행 실험을 하여야 하며, 때문에 많은 비용을 요구한다. 이에 실시간으로 제어계수를 조정하는 방법이 연구되고 있으나 최적화 알고리즘 자체가 많은 계산 비용을 요구하고 있어 UASs의 AFCs에 적용하기 어렵다.

본 논문에서는 LR과 보정학습방법(Reinforcement Learning)을 적용한 MLM을 제안하고 이를 통해 제어 계수 자동 조정을 수행하였다. 기체의 특성 중 무게를 한계치 이상으로 증가시킨 뒤 설계된 모델로 제어 계수를 자동 조정하여 시스템에 적용한 결과, 기존 수동 방식으로 조정하였을 때에 비하여 시간이나 비용적 측면에서 효율적인 것을 확인할 수 있었다. 또한 학습 알고리즘은 전체 시스템과 독립적으로 동작하므로 인터페이스만 맞춰주면 추후 더 효과적인 학습 알고리즘만 교체하는 것도 가능하다.

본 논문에서 제안한 MLM을 통해 자동으로 제어 계수의 조정이 가능하게 함으로써 기체의 크기나 무게와 같은 기체의 특성이 변화하더라도 α 에 능동적으로 대응할 수 있는 자동 비행 제어 시스템을 구축할 수 있다.

감사의 글

이 논문은 2009년도 한국항공대학교 교비지원 연구비에 의하여 지원된 연구의 결과임.

참 고 문 헌

[1] B. C. Kuo, Automatic Control System 5Edition, 1991.
 [2] L. N.Long, S. D. Hanford, O. Janrathitkarn, G. L. Sinsley and J. A. Miller, "A Review of Intelligent System Software for Autonomous," *Proc. of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Application*, 2007.
 [3] M. Pearce, R. Arkin and A. Ram, "The Learning of Reactive Control Parameters Through Genetic Algorithm," *Proc. of the 1992 IEEE/RSJ Intl. Conf. on Intelligent Robots and System, Raleigh, July*, 1992.
 [4] C. Ko, T. Lee, H. Fan and C. Wu, "Genetic Auto-Tuning And Rule Reduction of Fuzzy PID Controllers," *2006 IEEE Intl. Conf. on System, Man, and Cybernetics, October*. 2006.
 [5] Y. Abe, M. Konosho, J. Imai, R. Hasagawa, M. Watanabe and H. Kamiio, "PID Gain Tuning Method for Oil Refining Controller based on Neural Networks," *Proc. of the Second Intl. Conf. on Innovative Computing, Information and Control*, 2007.
 [6] C. Lin and C. S. George Lee, "Reinforcement Structure/Parameter Learning for Neural-Network-Based Fuzzy Logic Control System", *IEEE Trans. on Fuzzy System*, vol. 2, no. 1, February. 1994.
 [7] J. Lu, O. Ling and J. Zhang, "Lateral Control Law Design for Helicopter Using Radial Basis Function Neural Network," *Proc of the IEEE Intl. Conf. of Automation and Logistic, August*. 2007.
 [8] R. C. Nelson, Flight Stability and Automatic Control Second Edition, *TMcGraw-Hill higher Education*, 1998.
 [9] 장성욱, 이진걸, "The Self-Tuning PID Control Based on Real-Time Adaptive Learning Evolutionary Algorithm," *대한기계학회논문집*, 2008.
 [10] FDC 1.2 - A SIMULINK Toolbox for Flight

Dynamics and Control Analysis

[11] I. H. Witten and E. Frand, DATA MINING Practical Machine Learning Tools and Techniques Second Edition, Morgan Kaufmann Publishers, 2005.
 [12] D. C. Montgomery, E. A. Pech and G. G. Vining, Introduction to Linear Regresson Analysis Fourth Edition, *Wiley Interscience*, 2006.
 [13] S. Russell and P. Norvig, Artificial Intelligence A Modern Approach Second Edition, *Prentice Hall*, 2002.

문 미 선 (文美善)



2004년 2월 : 한국항공대학교
전자정보통신공학부(공학사)
2006년 2월 : 한국항공대학교
컴퓨터공학과(공학석사)
2006년 3월~현재 : 한국항공대학교
컴퓨터공학과(박사과정)
관심분야 : 항공시스템소프트웨어,
기계학습, 지능시스템

송 강 (宋壩)



2008년 2월 : 한국항공대학교
전자정보통신공학부(공학사)
2010년 8월 : 한국항공대학교
컴퓨터공학과(공학석사)
관심분야 : 항공시스템소프트웨어,
AHRS

송 동 호 (宋東鎬)



1991년 5월 : 영국 Newcastle대학교
전산학과(박사)
1995년 3월 ~현재: 한국항공대학교
항공전자정보통신
공학부 교수
관심분야 : 항공시스템소프트웨어,
Fault Tolerant, OS