Learning to Fly: Modeling Human Control Strategies in an Aerial Vehicle

Stephen B. Stancliff
stanclif@mil.ufl.edu

Michael C. Nechyba
nechyba@mil.ufl.edu

Machine Intelligence Laboratory
Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611-6200

Abstract

Much work has been done in recent years to abstract computational models of human control strategy (HCS) that are capable of accurately emulating dynamic human control behaviors. Land-based autonomous vehicles, both in simulation and on real roads, have made successful use of this modeling formalism. Little work has been done, however, in attempting such skill transfer from humans to aerial robotic vehicles. Although control of an aerial vehicle is quite different from that of ground vehicles, we contend that human pilots can potentially serve as excellent guides in the development of intelligent autonomous aerial vehicles. As a first step in modeling human control strategies in aerial vehicles, we are developing a robotic airplane (Figure 1) as an experimental platform for studying human-to-machine skill transfer in aerial vehicles. This paper describes the configuration of this airplane, the results of early experiments, and future planned experiments.

Figure 1 - The Avigator platform

1. Introduction

Motivation

Over the past two decades, rapid advances in computer performance have not been matched by similar advances in the development of intelligent robots. Humans are much better at performing complex dynamic skills than at describing those skills in an algorithmic, machine-codeable way. This has limited our ability to develop intelligence in robots and other machines. This inability has limited not only the capabilities of individual machines, but also the extent to which humans and robots can safely interact and work cooperatively. There exists a profound need to abstract human skills into computational models which are capable of realistic emulation of dynamic human behaviors.

Autonomous control and navigation of ground vehicles is one area of robotics research which has benefited from learning through observation of humans. Pomerleau [1,2], for example, has implemented in the ALVINN system real-time road-following using data collected from a human driver. A static feedforward neural network with a single hidden layer learned to map from coarsely digitized images of the road ahead to a desired steering angle. The ALVINN system has been demonstrated successfully at speeds up to 70 mph. Pentland and Liu [3] have applied hidden Markov models to inferring a driver's high-level intentions, such as turning and stopping. Finally, [4,5,6] address the autonomous control of a dynamic car -- including steering and acceleration -- through observation and modeling of human driving using a driving simulator.

Surprisingly, little work has been done in using observation of human pilots to create intelligent autonomous aerial vehicles. An intelligent autonomous aerial vehicle could have application in a number of areas. Many of the activities that currently involve remotely piloted vehicles (RPVs) would benefit in some way from automation. In many applications, such as surveying, reconnaissance, and target acquisition, it may be possible to automate the entire mission. For other applications, where more sophisticated control is required, it may be that a human pilot is still required, but that the less complex parts of the mission can be automated. Adding intelligence to RPVs could reduce the amount of skill required of the human pilots, and could also allow one pilot to control multiple vehicles.

Although the control challenges in flying are different from those in driving, the basic paradigm of learning from humans is equally applicable. We are therefore attempting to extend to aerial vehicles some of the methods previously used for learning in ground vehicles. As a first step in this process, we are developing a robotic airplane as an experimental platform.

Project outline

We have divided the overall project into three stages according to the level of autonomous behavior to be achieved: (1) The airplane will be able to fly straight and level, maintaining a given heading, and will be able to make a turn to a new heading; (2) The airplane will be able to land and take off; (3) The airplane will be able to navigate to given map coordinates.

At this point, we have assembled and tested the hardware necessary to implement stage one, and have shown that we can produce an accurate model of the human pilot for straight and level flight. In the near future we will test this model by letting the computer fly the airplane.

2. Platform Description

Mechanics

The basis of the platform is a radio-control (R/C) airplane kit, the Sig *Kadet Senior*. This airframe was chosen because of its slow and stable flying characteristics and because it has a large payload capacity, both in volume and in weight. The airframe has a wing area of 1150 in^2, and an empty weight of about 6 lbs, giving a wing loading of about 12 oz/ft^2. An R/C airplane is still considered 'light' at 20 oz/ft^2, and we should be able to easily go to 30 oz/ft^2 with no significant detrimental effects on the flying characteristics. A wing loading of 30 oz/ft^2 corresponds to a payload of about 9 lb. In reality, we will run out of internal volume before we reach 9 lb of electronics, although the fuselage could be modified to provide more volume.

During construction, the airframe was modified in many small ways to accommodate the needs of this project. In many places thinner wood was substituted in order to reduce weight. The interior of the fuselage beneath the wing was modified in order to maximize the volume of the payload compartment. An extra servo was used for the nosewheel, in order to eliminate the need for control-rods running through the payload compartment. The forward servos were moved into the nose, and the rearward servos placed at the very rear of the payload compartment, again to maximize payload volume and accessibility.

When the platform was first designed, it was to be propelled electrically. Electrical propulsion is desirable for a project like this because an internal combustion engine has traits that are harmful to sensitive electronics, specifically vibration and caustic fluids. We designed a propulsion system consisting of a brushless motor, a reducing gearbox, and nickel-metal-hydride batteries [7]. This system was tested extensively, and provided the capability to lift a 3-4 lb payload for flight durations of 8-10 minutes, which was adequate performance to accomplish stages 1 and 2 of the project. Unfortunately, the electric drive system proved too fragile to meet our needs in the areas of consistency and reliability. After many trips to the repair shop, we decided to rebuild the platform with an internal combustion engine.

Most R/C aircraft use small two-cycle engines. In recent years, four-cycle engines have become more popular due to their lower noise, lower vibration, and more reliable and consistent performance. The downside of a four-cycle engine is that it produces less power per pound and much less power per dollar in comparison to a two-cycle engine. Reducing vibration was a high priority for us, so we chose to install a four-cycle engine. The engine we are using has a displacement of 0.52 in^3 and is rated at 0.9 hp.

The performance of the airplane was significantly improved with the new engine -- although the takeoff weight dropped by less than a pound compared with the electric, the takeoff run required on a calm day with a 2 lb payload went from about 50 ft to about 30 ft. This performance was not noticeably changed when we tested it with an additional 5 lb of payload.

The new engine brought many difficulties in the form of vibration. The sensors and the computers had been hard mounted with the electric motor, and performed well. With the new engine, several components showed erratic behavior. Several methods of soft-mounting the components were tried. For most of the components, a sandwich of soft foam between plywood mounts provided adequate damping. For the computer's hard drive and the accelerometer-based tilt sensor, however, no adequate vibration reduction method has yet been found. As a result, we have had to leave off the hard drive and rely on a small solid-state drive, and also leave off the tilt sensor and rely on the tilt sensor built into the compass.

The human-in-control system is provided by a standard six-channel R/C radio system and standard R/C servos. Four channels are used for control of the airplane -- rudder, elevator, nosewheel, and throttle. A fifth channel, which is a toggle switch, is used to switch between human and computer control of the aircraft. The sixth channel is currently unused and may be used in the future to send commands to the computer from the ground. The only modification to the R/C system has been tapping into the receiver to bring the servo-control signals out to the computer.

Intelligent Electronics

The primary processing power of the platform is provided by a 386-class processor in a PC/104 format. This PC/104 board provides all of the features usually found on a desktop computer, including video and ethernet, on a 3.6" by 3.8" board, and weighs less than 4 oz. The PC/104 bus makes future expansion of the system straightforward, using readily-available daughter-boards. A 386 was chosen over a 486 or 586, in large part because it uses much less power, and also because of its lower cost.

In addition to the primary computer, there is a Motorola MC68HC11 (HC11) microcontroller on a Machine Intelligence Laboratory (MIL)-designed MRC-11 board. This board provides the HC11 in expanded mode with 32k of SRAM and 32k of ROM, and provides convenient access to all of the HC11 pins via a header.

The HC11 is responsible for capture of all sensor and control data except for the compass data, which goes directly to the 386. The HC11 transmits this information to the 386 through an RS-232 serial connection. It also accepts control commands from the 386 and then produces the control signals for the servos.

Data storage is provided by an 0.7 MB solid-state disk on the PC/104 board. This is adequate for storage of about 10 minutes of data. We have ordered a 64 MB solid-state disk which will enable us to store many flights worth of data onboard. This will be a significant improvement since we currently have to remove the wing between flights in order to copy the data off of the 386 onto a laptop computer or a floppy disk.

We are currently using MS-DOS as our operating system, due to the limited storage available. When we were using a hard drive with the electric propulsion system, we experimented with Linux and found it to have many advantages over MS-DOS including (1) a higher sampling rate and (2) the ability to log in from another computer and debug programs while they are running. The major disadvantage of Linux, besides its space requirements, is that it takes a very long time to boot on a 386. When we install the new solid-state drive we will experiment with some new versions of Linux that are designed for embedded applications.

The sensor suite currently provides heading, pitch, and roll data, all of which are provided by a Precision Navigation electronic compass, which interfaces to the 386 through an RS-232 interface. The tilt sensor in this compass is not very good for our application, since it is of the liquid-level type and thus ill-suited to dynamic environments. It also has a limited tilt range of ±45°.

We have developed a tilt sensor around the Analog Devices ADXL202 accelerometer, which has superior performance to the compass' tilt sensor, but which we are currently unable to use because it is very sensitive to the engine vibrations.

Additional sensors are under development, including a pressure-based altimeter for coarse altitude measurement, an ultrasonic ranger-based altimeter for finer altitude measurement near the ground, and a pressure-based airspeed sensor.

Tying the intelligent components into the radio control system is a special multiplexing circuit which was designed to allow the human pilot to switch between human control and computer control of the aircraft using an extra channel on the radio. This circuit operates off of the receiver battery, and is independent of the computer systems so that, in the event of a computer failure, the pilot can still assume control of the aircraft.

Software

During the data-collection phase of the project, the HC11 is responsible for capturing sensor inputs, and also for decoding the pulse-width-modulated (PWM) signals from the radio receiver. All of this data is then transmitted through the serial interface to the 386.

During data collection the function of the 386 is to collect the serial data from the HC11 and the compass, timestamp them, and save them to disk. This raw data consists of five numbers representing heading, pitch, roll, rudder control position, and elevator control position.

Data processing and training are performed on the ground using more-powerful computers. The first step in data processing is to manually select sections of the data which are representative of the behavior to be learned (e.g., 'flying straight and level'). This is a subjective task, since for dynamic real-world data it is often hard to say exactly where 'straight' stops and 'turn' starts.

After several data segments have been selected, the data are preprocessed by the following steps: (1) heading values are adjusted by increments of 360° to eliminate the problem of values flipping across the 360°=0° boundary. (2) Change-in-heading (Δh) values are calculated, and any value of $|\Delta h|$ greater than 45° is replaced with 0, since it is obviously an invalid reading (there are several possible causes of this situation, but the most likely is data dropout). (3) The data is resampled to a constant timestep of 0.1 s. The raw data, even without dropouts, have varying timesteps, due to the lack of synchronization of the two serial streams, and also due to some qualities of the solid-state disk. Any time period of less than 300 ms is considered a valid interval, and the data is resampled. A gap of greater than 300 ms is considered a break in the data stream and the data following the gap is flagged as a new section of data. (4) the data are scaled to lie within the range ±1. (5) Finally, the data are reformatted as a time-history so that a set of data now consists of 20 values, 18 inputs (current and past sensor values and past control values) and 2 outputs (current control values).

Data preprocessing steps 1 and 2 are performed more or less by hand using a spreadsheet, and steps 3-5 are performed by a C program. (All code is written in C).

After preprocessing, the data is used to train a cascade neural network (CNN). We have previously found that continuous human control can be modeled well through CNNs, which are powerful nonlinear function approximators offering several advantages over more traditional neural network architectures: (1) the network architecture is not fixed prior to learning, but rather adapts as a function of learning [8]; (2) hidden units in the neural network can assume variable activation functions [9]; and (3) the weights in the neural network are trained through the fast-converging node-decoupled extended Kalman filter [9]. The flexibility of these cascade networks is ideal for HCS modeling, since few *a priori* assumptions are made about the underlying structure of the human controller.

In the execution stage, the trained neural network is used by the 386 to produce control outputs from the sensor inputs. These control values are passed to the HC11, where they are used to generate appropriate PWM signals which are then sent to the control servos. We are currently in the process of shaking down the hardware required to attempt this step.

3. Experimental Results

To date we have performed neural network training using data representing straight and level flight. For this data the result of the neural network training has been a purely linear controller. This is unsurprising, given the highly linear correlation seen between the control inputs and sensor outputs. Figure 2 shows the correlation between the rudder control input and the roll angle of the airplane.

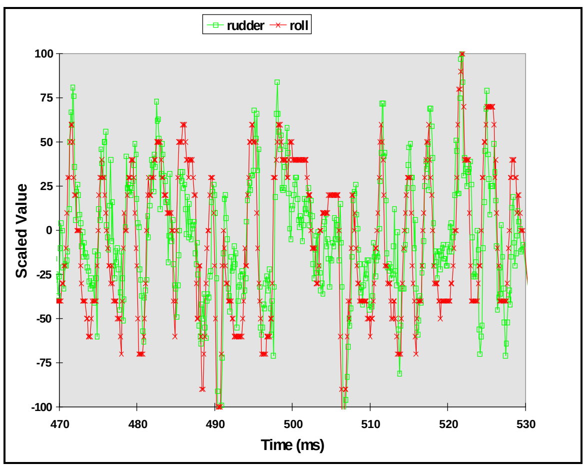

Figure 2 - Rudder and roll angle

The resulting control system was then used to calculate predicted values for pilot commands at time t given the time-history of sensor and control values. Our control system was found to predict the pilot's control actions very well, with RMS error less than 0.1%. Figure 3 illustrates the close correlation between the predicted and actual control values.

4. Conclusion and future work

This paper describes the first stages in the development of an autonomous airplane using human skill modeling. At the time of this writing, we have shown that we can generate a model which accurately predicts the next pilot command from past commands and current and past sensor data. In the near future, we expect to test the model by allowing the computer to fly the airplane in straight-and-level flight. From that point we will work to develop additional models of turning, landing, and takeoff behaviors.

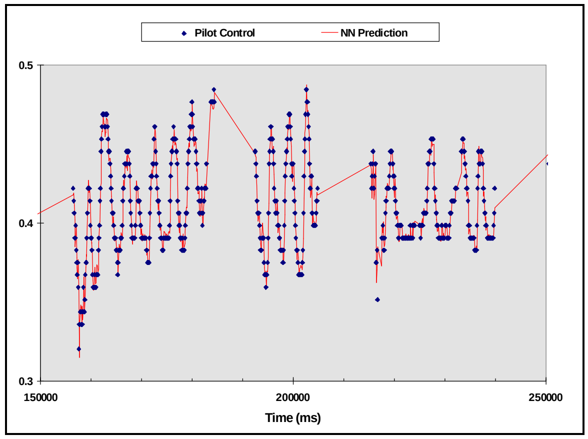

Figure 3 - Predicted and actual controls

Acknowledgments

We would like to acknowledge the support of the following people and companies, without whom this project would not have been possible: (1) the members of the Machine Intelligence Laboratory; (2) Motorola, a major sponsor of the MIL; (3) EMJ Embedded Systems, which provided a discount on the purchase of the PC/104 computer and the solid state disk; (4) Eveready, which provides batteries to the MIL; and (5) Analog Devices, which provided the ADXL202 accelerometer chips.

References

[1] D.A. Pomerleau, "Neural Network Perception for Mobile Robot Guidance," Ph.D. thesis, Carnegie Mellon University, 1992.

[2] D.A. Pomerleau, "Reliability Estimation for Neural Network Based Autonomous Driving," *Robotics and Autonomous Systems*, vol. 12, no. 3-4, pp. 113-9, 1994.

[3] A. Pentland and A. Liu, "Toward Augmented Control Systems," *Proc. Intelligent Vehicles*, vol. 1, pp. 350-5, 1995.

[4] M.C. Nechyba and Y. Xu, "Human Control Strategy: Abstraction, Verification and Replication," *IEEE Control Systems Magazine*, vol. 17, no. 5, pp. 48-61, 1997.

[5] M.C. Nechyba, *Learning and Validation of Human Control Strategies*, Ph.D. thesis, Carnegie Mellon University, 1998.

[6] M.C. Nechyba and Y. Xu, "On Discontinuous Human Control Strategies," *Proc. IEEE Int. Conference on Robotics and Automation*, vol. 3, pp. 2237-43, 1998.

[7] S. Stancliff, J. Laine, and M. Nechyba, "Learning to Fly: Design and Construction of an Autonomous Airplane," *1999 Florida Conference on Recent Advances in Robotics*. (http://cimar.me.ufl.edu/FLA99/session4/flying.pdf)

[8] S.E. Fahlman, L.D. Baker and J.A. Boyan, "The Cascade 2 Learning Architecture," Technical Report CMU-CS-TR-96-184, Carnegie Mellon University, 1996.

[9] M.C. Nechyba and Y. Xu, "Cascade Neural Networks with Node-Decoupled Extended Kalman Filtering," *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, vol. 1, pp. 214-9, 1997.