Autonomous Flight Control for an RC Helicopter

Michael Camilleri
Department of Communications and Computer Engineering
University of Malta
Msida, Malta

Dr. Kenneth Scerri
Department of Systems and Control Engineering
University of Malta
Msida, Malta

Dr. Ing. Saviour Zammit
Department of Communications and Computer Engineering
University of Malta
Msida, Malta

Abstract – **This paper presents an autopilot system for a Radio Controlled (RC) helicopter, in a simulation environment. The controller is designed using Fuzzy Logic in order to achieve simplicity and cost effectiveness. The system controls the helicopter from takeoff to landing, allowing the craft to reach any position and hover stably with a defined yaw attitude at the request of the user. The Fuzzy Inference Systems (FIS) in use were able to allow for these operations with more than satisfactory results, even in simulated winds.**

Index Terms — **Autonomous Helicopter Flight, fuzzy logic control, Unmanned Aerial Vehicles.**

I. INTRODUCTION

Although helicopters are some of the most agile (and thus useful) machines ever invented, they remain some of the most difficult contraptions to fly and control, with autopilot technology still in its infancy. This stems in part from the complexities involved in mathematically modeling helicopter dynamics [1], which are somewhat unstable and highly coupled [2] making it very difficult to control them.

Some projects have indeed been implemented in this area ([3], [4], [5], [6] and [7]). However most, if not all, involve elaborate solutions which are too complex and expensive to be applicable to everyday situations; many are oriented towards military applications. This project aims to achieve adequate flying performance for a Radio Controlled (RC) helicopter with a focus on simplicity of design and cost effectiveness. Fuzzy Logic Controllers (FLCs) are ideal for these goals: they are certainly easier to design, requiring limited insight into helicopter mathematics, while allowing for relatively cheap hardware for a physical implementation. In fact such techniques have already been tested, as highlighted in [8], [9] and [10]. This project builds upon such systems to develop a controller which is easy to design without having to go into complex state-space representation.

The project uses a model of the Honey Bee King II RC helicopter [14], in SVK Systems' Clearview Simulator [16]. The user interacts with the controller by commanding a set of positions and yaw attitudes, which the autopilot attempts to achieve. The control system is modularised according to each of the control axes: collective for height control, lateral and longitudinal cyclic for horizontal motion and heading control through the tail collective. The Fuzzy Logic Controllers were designed using Matlab®'s Fuzzy Logic Toolbox, and ported to the C++ application implementing the controller as Look-Up Tables (LUT). The latter application interfaces with the simulator using TCP/IP.

The rest of the paper is organized as follows. Section II introduces the theory regarding the kinematics involved in the project: section III briefly covers the software architecture, following with a description of the control methodology used, in section IV. Section V presents the results achieved. Finally the paper concludes with an evaluation of the same results, together with a brief discussion of future work in this area.

II. BASIC KINEMATIC MODELLING

The helicopter is represented in 3D space by its position (X, Y and Z co-ordinates) and its rotations about the co-ordinate axis (yaw, pitch and roll). Due to the six-degree of freedom dynamics (6DOF), three frames of reference are used. The *earth* co-ordinate system is centred at the starting location of the helicopter, and is used to map positions of the helicopter in 3D space. The *model* axis is positioned at the centre of gravity (c.o.g.) of the helicopter, and is used to define rotations with respect to the initial orientation. This can be conceptualised as having another co-ordinate axis (*spatial* frame) at the c.o.g. of the helicopter but with its orientation fixed as for the earth frame. Rotations are then defined between the model frame and the latter one.

In order to use conventional matrix transformations, the well-known North-East-Down (NED) co-ordinate system is employed [11]. However, the simulator itself uses a different representation, with the model facing –X, the left side pointing towards +Z and the pilot head towards +Y. Thus, in retrieving the position/velocity data from the simulator, these had to be appropriately converted. Acceleration data is obtained by a first order derivative of the velocity data, utilising the current and previous sample values:

$$Acceleration_n = \frac{Velocity_n - Velocity_{n-1}}{elapsed\ time} \quad (1)$$

Additionally, a moving average of the acceleration data is then computed, covering ten sample points, to smooth out any noise. This size appeared to be the best at the sampling rate used, in reducing noise, while not contaminating the acceleration data with values that are too old.

With respect to orientation, the simulator provides only the rotational velocities in model co-ordinates. In order to get at the attitude, these must be converted into the spatial frame (as indicated in [12]) and integrated over the elapsed simulation time. Angular acceleration is calculated in a similar way to linear acceleration. In this way, angular rates and accelerations are expressed in the model frame which is adequate, since the control systems act in the same frame.

A further point needs to be noted regarding the angle values. Obviously, orientation is limited over a range of 2π, either between 0 and 2π or between –π and +π. As such, care must be taken in the integration process to normalise the angle to these regions as described hereunder:

$$Angle_{Norm} = \begin{cases} Angle_{In} - 2\pi & when\ Angle_{In} > \pi \\ Angle_{In} + 2\pi & when\ Angle_{In} < -\pi \\ Angle_{In} & otherwise \end{cases} \quad (2)$$

Also note that all the Fuzzy Inference Systems (FIS) act upon an error in some axis of translation/rotation, defined as:

$$Error\_Value = Target\_Value - Current\_Value \quad (3)$$

In order to start off at a known state, a calibration routine is implemented before the controller itself starts. In this routine, calibration data regarding the current position is recorded and used to calculate offsets to decrease from the subsequent readings so that it is as if the model and world co-ordinate systems are initially aligned.

III. SOFTWARE ARCHITECTURE

The purpose of the controller is to read telemetry data given by the simulator, determine the control outputs to pass back to the same simulator and log results to secondary storage. Four main classes are implemented, depicted graphically in Figure 1.

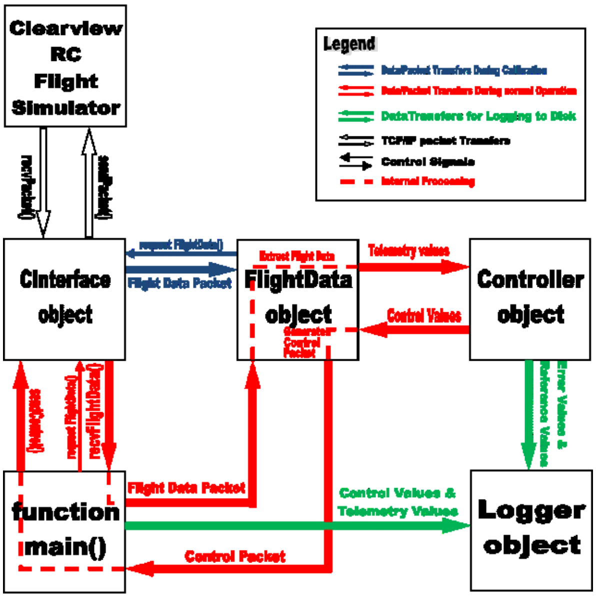

Fig. 1. Overview of the Software Architecture

The *CInterface* object implements the TCP/IP socket for communicating with the simulator through strings of characters. A *FlightData* object handles the storage and interpretation of the simulator-generated telemetry. The control algorithm itself is implemented within the *Controller* object. This class accepts telemetry information from the *FlightData* object and determines the control outputs to vary to achieve the user's requested positions/attitudes. It implements the separate controllers for each axis as a set of multi-dimensional LUT's, for efficiency, as well as the main *Autopilot* Finite State Machine (FSM). A further *Logger* object is implemented to abstract the process of logging data to file for later retrieval. These are: telemetry, control values and reference/error information, all appropriately time-stamped. Logging is done via binary files, since this was found to be the most time-efficient method as opposed to text files or C++ data structures (such as lists and vectors). These are then later analysed and plotted using Matlab®.

IV. CONTROLLER DESIGN

The objective of the project is to control the helicopter's absolute position in space. A human pilot is only required to enter x-y-z co-ordinates, together with a yaw attitude, and the controller attempts to achieve a stable hover at this position. Although helicopter dynamics are severely coupled, this is most evident in aggressive manoeuvres [13]. For simple flight patterns such as the controlled hover required in this project, the dynamics can be assumed to be decoupled. This is aided by the speed at which the control inputs are generated (333Hz in the final setup), allowing the controllers to quickly compensate for coupling through the refresh rate of telemetry inputs and control commands, a fact verified by simulation results. This allowed the control problem to be divided according to the control axis, and a specific controller designed independently for each.

Five FLCs were designed. The *Height* FIS maintains the necessary vertical position by varying the collective input, while the *Yaw* FIS achieves a desired heading using the tail collective. A separate *Landing* controller takes over from the *Height* FIS during landing operations, with the emphasis being on a smooth descent to avoid crashing. X-Y position is controlled through a cascade of two systems: a *Hover* controller which outputs pitch/roll angles in order to correct for horizontal errors and a *Cyclic* FIS which attempts to achieve these attitudes through control of the cyclic input.

Each of the fuzzy controllers receives feedback from the simulator in the form of position/angle error, velocity, and in all but the *Yaw* and *Hover* FIS, acceleration. Velocity and acceleration feedback serve to limit the velocity near the target position/angle to a reasonable value in an attempt to reduce overshoots, without suffering in response time. The details of each controller are listed in Table I. In turn, these individual FIS act under the overall control of the *Autopilot* FSM. Its role is to switch on/off each controller depending on the current situation and user commands. The choice of using FIS was based on several factors, foremost being simplicity of design. Moreover, since the rules are developed in an intuitive fashion, the controller is generalised and potentially applicable to different models with minor tuning. In addition, whereas mathematical models usually fail to capture all the subtleties of helicopter dynamics (often a system must be linearized or simplified), the rule-base for an FIS can incorporate any non-linearities which are intuitively deduced. Using FLCs also simplified design, for example by exploiting the similarities between the pitch and roll axis.

A major design decision was whether to use incremental or absolute outputs: i.e. whether the FLCs output absolute control values or changes with respect to the previous ones. Following appropriate testing, the *Height*, *Landing* and *Yaw* FIS were implemented using incremental outputs, while *Cyclic* and *Hover* control use the absolute methodology.

TABLE I OVERVIEW OF THE FIS SYSTEMS OF THE CONTROL ALGORITHM*

Name	**Controls…**	**Input1**	**Input2**	**Input3**	**Output**
HEIGHT (57 Rules)	Height, during take-off and hovering	Height Error 7 MF ±3m	Vertical Velocity 5 MF ±7m/s	Vertical Acc. 5 MF $\pm 15 m/s^{2}$	Collective Incremental 9 MF ±0.02
LANDING (16 Rules)	Vertical, position for landing operations	Height 4 MF 0/10m	Vertical Velocity 5 MF ±7m/s	-	Collective Incremental 7 MF ±0.02
YAW (35 Rules)	Heading, through Tail Collective	Yaw Error 7 MF $\pm 3^{c}$	Yaw Velocity 5 MF $\pm 10^{c}/s$	-	Tail Pitch Incremental 7 MF ±0.05
CYCLIC (58 Rules)	Pitch/Roll Attitude	Angle Error 7 MF $\pm 0.5^{c}$	Angular Velocity 7 MF $\pm 2^{c}/s$	Angle Acc. 3 MF $\pm 20^{c}/s^{2}$	Cyclic Absolute 9 MF ±0.8
HOVER (28 Rules)	X/ Y position through CYCLIC commands	Horiz. Error 7 MF ±5m	Horiz. Velocity 5 MF ±5m/s	-	Angle Absolute 7 MF $\pm 0.1^{c}$

*The values in the Input/Output columns refer to the number of Member Functions (MFs) with the range of the fuzzy variables underneath them. Any dimensionless values in the Output refer to helicopter control inputs which are limited to ±1.0 full scale.

All FIS use Mamdami Logic, with rules based on the 'and' method and with a centroid of area defuzzification function. The ranges for the input variables were chosen after tests to determine the maximum dynamic limits of the helicopter model. The membership functions (MF) are usually trapezoidal at the edges, with the exception of the sigmoid for the intermediary ones for a smoother transition.

Some important features of each FIS should be noted. The nine MF's for the output of the *Height* controller allow for very minute and precise changes. Also, the rule-base gives preference to upward thrust to account for the effect of gravity. In the case of the *Yaw* controller, the error input is normalised to between $\pm\pi$ so that if the system overshoots the limit, the controller acts on the shortest distance. For the *Cyclic* controller the positional errors and velocity are rotated into the model co-ordinate system, according to [12].

In addition, it was necessary to augment the *Hover* FIS with two integral schemes. The first is enabled when the helicopter achieves steady state. It accumulates the error in the X (or Y), multiplied by an integral control factor (0.0001 for the X and 0.0005 for the Y, due to the roll axis having slightly slower dynamics), and adds it to the error input of the same FIS. The other scheme adds directly to the output of the *Cyclic* FIS, and was necessary in windy conditions, since the limits set at $\pm 0.1^c$, although sufficient to provide stability, were unable to overcome wind disturbances. In this case, enabled by the user, the integral scheme accumulates the error multiplied by 0.001 and saturated to ±0.25 only if the error is constant or increasing, and decays by a forgetting factor of 0.9 otherwise. This was done to maintain stability.

The Autopilot FSM brings the individual controllers together, feeding the necessary inputs to each and activating certain aspects on or off as needed. The helicopter starts off in GROUNDED mode, when all control values are at the neutral position. On takeoff command, the controller starts to correct for altitude and yaw attitude, while maintaining the helicopter horizontal. No correction for X/Y errors is performed to avoid coupling effects with the collective controller. Once the desired height is reached, the HOVER status kicks in and the autopilot activates the *Hover* FIS together with the associated integral scheme(s) in order to correct for X/Y positions. The procedure is repeated on every call to takeoff. Once the landing command is given, the FSM transitions to the LAND state, at which point the *Hover* FIS continues to maintain X/Y position (integral control is shut off however) while the *Landing* FIS is now in control of the helicopter, as it brings it down to ground. At this point, the FSM reverts back to the GROUNDED state.

V. TESTING AND RESULTS

Testing was carried out entirely in the simulation environment, and in a modular fashion. The helicopter was instructed to transition to some reference position/attitude (keeping all else constant) and the respective controller attempted to achieve it. During the simulation, data was logged to file, and analysed afterwards to fine-tune the FIS being tested. Finally, the whole control system was brought together in an entire 'mission' from take-off, through a range of set-points and finally landing back to ground.

Altitude control was the first axis to be designed. Testing results yielded indications as to what the control frequency should be (200Hz, and later increased to 333Hz), together with the need for acceleration feedback, without which the height suffered from sustained oscillations. Otherwise, the system performed admirably with an overshoot of about 20% and no more than one oscillation before settling down. The *Yaw* response was the best of the axes controlled, achieving minimal errors of less than 2%, within 1 second of activation, and with minimal oscillations of less than 3%.

Cyclic control was developed mainly for the pitch axis, and then ported out to the roll axis as well. This was done in order to reduce design time, taking advantage of the similarities in the dynamics of the two axes. However, in the final implementation, certain features were individually fine-tuned. Very satisfactory resuls were achieved, both for the pitch and roll axes, as shown in Figs. 2 and 3 respectively.

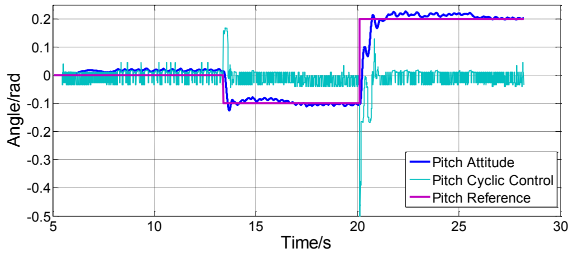

Fig. 2. Pitch Reference and Attitude, with cyclic commands

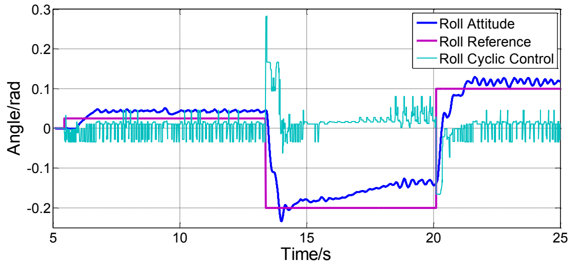

Fig. 3. Roll Reference and Attitude, with cyclic commands

The helicopter control system was brought together in a full set-point tracking example, tabulated below (Table II) and the results depicted in Figs. 4 through 7. Table III, shows some statistics as to the steady state hovering errors. The results indicate the ability of the FIS coupled with the integral control scheme to correct for errors and maintain a stable hover with minimal steady state errors in all axes.

TABLE II SET-POINTS COMMANDED WITH TIME STAMP

X-Pos/m	Y-Pos/m	Z-Pos/m	Yaw/^{c}	Time/s
0.0	0.0	-5.0	0.0	006.9
0.0	3.0	-7.0	1.0	051.2
10.0	3.0	-10.0	0.0	080.4
10.0	6.0	-10.0	-0.5^c	124.7s
10.0	6.0	0.0	-0.5^c	144.3s*

*this is the landing command.

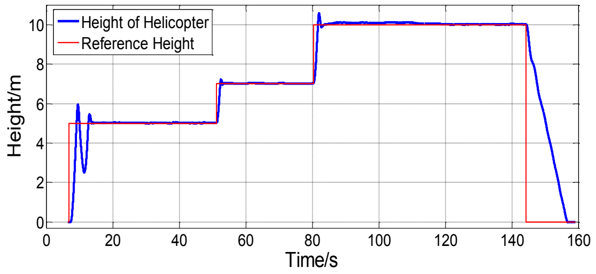

Fig. 4. Height of Helicopter

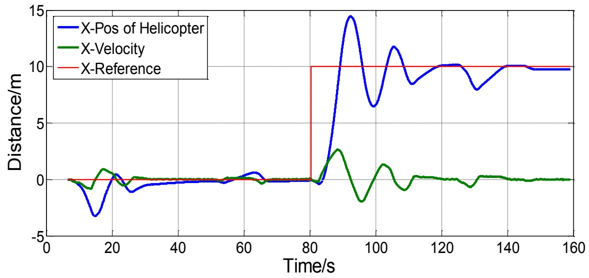

Fig. 5. X-Position of Helicopter with velocity

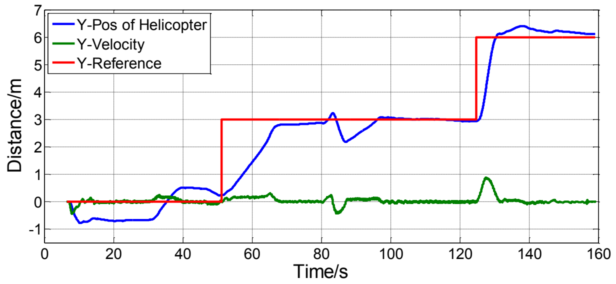

Fig. 6. Y-Pos of Helicopter with velocity

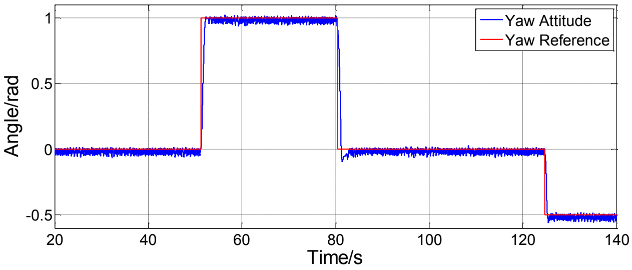

Fig. 7. Yaw Attitude of Helicopter with reference values

TABLE III STATISTICS FOR STEADY STATE ERRORS IN HOVER

Axis	**Min**	**Max**	**Mean (abs)**	**Std Dev (abs)**
X/m	8.85e-5	0.7929	0.2506	0.1579
Y/m	3.47e-8	0.3204	0.0795	0.0779
Z/m	1.91e-6	0.0574	0.0189	0.0112
Yaw/rad	1.46e-8	0.0947	0.0217	0.0163
Pitch/rad	4.73e-7	0.0558	0.0175	0.0097
Roll/rad	9.00e-7	0.0568	0.0172	0.0093

The ability of the controller was also tested under windy conditions. The height controller managed to keep the helicopter stable with 5m/s up/down drafts and at 100% turbulence (fluctuations in this speed), although X/Y control suffered. The additional integral control scheme was able to achieve very good stability and steady state errors of less than 0.5m with winds of 5m/s (Beaufort Force 3).

VI. CONCLUSION

The above results show that the aims of the project were largely achieved. A Fuzzy Logic Controller was designed, capable of controlling a helicopter to achieve user-defined positions and able to autonomously take-off, hover and land at the user's demands. Considering that often the helicopter was required to perform changes in various DOF's (i.e. position and yaw attitude) simultaneously, the response and stability maintained is more than satisfactory. Furthermore, it must be kept in mind that the rules were developed by a non-expert in the field, a testament to the flexibility and ease of design of the proposed control scheme. The helicopter responded very well even under 5m/s winds, which is notable considering the lightweight nature of RC helicopters [15].

Future work would concentrate on improving the transient phase of the helicopter, to avoid oscillations which would render it difficult to manoeuvre in tight spaces. Some degree of mathematical modelling to account for the axial coupling can aid in this regard. The end goal would be to physically implement the system. Further studies would be needed in this case, mainly to obtain the same feedback signals as in simulation. Usually, the main source of telemetry would be an inertial measurement unit composed of a three-axis accelerometer (measuring linear acceleration) and a three-axis angular rate sensor (for angular velocity). These would be transformed and integrated as necessary to get at the position and attitude data: however, appropriate filtering would be needed to reduce the effects of drifts and noise which present a formidable problem. It is envisaged that for precise positional control, further sensors such as GPS sensors and ultrasonic range finders would need to be incorporated. Although other techniques such as vision or laser guidance are also possible, these would offset the original goal for a low cost design.

VII. REFERENCES

[1] D. Krol, M. Lower and B. Szlachetko, "Helicopter Intelligence in Hover: Quality Improvement of the Fuzzy Regulator", in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, 2006, vol.1 pp. 326-331.

[2] Gareth D. Padfield. *Helicopter Flight Dynamics: The theory and application of flying qualities and simulation modelling*. Oxford, Blackwell, 1996, pp 16-36, 437.

[3] M.N. Krasilshchikov, K.I. Sypalo, D.A. Kozorez, "Helicopter Terrain-Following Integrated Navigation & Control System based on GPS/GLONASS Technique and Technology", in *Proceedings of the European Conference for Aerospace Sciences, Moscow* 2005.

[4] Alberto Isidori, Lorenzo Marconi and Andrea Serrani, "Robust Nonlinear Motion Control of a Helicopter", *IEEE Transactions On Automatic Control*, vol. 48, no.3, pp. 413-426 March 2003.

[5] Eric N. Johnson and Suresh K. Kannan, "Adaptive Flight Control for an Autonomous Unmanned Helicopter", in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, California, August 2002.

[6] R.L.Barron, R.L.Cellucci, P.R.Jordan, N.E.Bearn, P.Hess and A.R.Barron, "Applications of Polynomial Neural Networks to FDIE and Reconfigurable Flight Control", in *Proceedings of the IEEE National Aerospace and Electronics Conference,* May 1990, pp 507-519 vol.2.

[7] K. Harbick, J. F. Montgomery and G. S. Sukhatme, "Planar Spline Trajectory Following for an Autonomous Helicopter", in *Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2001, pp. 408-413.

[8] Dimitrios Iakovou "*Fuzzy Control for Helicopter Aviation*", M.Sc Project (for Individual module) University of Twente, Netherlands, June 2002.

[9] M. Lower, B. Szlachetko and D. Krol, "Fuzzy Flight Control System for Helicopter Intelligence in Hover", in *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, 2005, pp. 370-374.

[10] Richard D. Garcia, "*Designing an Autonomous Helicopter Testbed: From Conception through Implementation*", Ph.D. Thesis, University of South Florida, Florida, January 2008.

[11] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd Edition., John Wiley & Sons, Hopoken, NJ, USA, 2003.

[12] L. Chetcuti Zammit, "Simulation of Flight Control for an Autonomous Helicopter", B.Eng. dissertation, University of Malta, Malta, 2010.

[13] Richard D. Garcia, "Designing an Autonomous Helicopter Testbed: From Conception through Implementation", Ph.D. Thesis, University of South Florida, Florida, January 2008.

[14] Hobby-estore (2011), Esky Honey Bee King II 6CH CCPM Electric RC Helicopter RTF. [Online]. Available: www.hobby-estore.com/Esky-Honey-Bee-King-II-6-CH-Electric-RC-Helicopter-p/eh-ek-e016.htm

[15] J. Salt (2011), RC Helicopter Fun. [Online]. Available: www.rchelicopterfun.com.

[16] SVK Systems (2011), Clearview RC Flight Simulator. [Online]. Available: http://rcflightsim.com/