Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Flight Controller and Low-Cost Test Environment for a Simulated Helicopter

Martin C. Terblanche*, Kenneth R. Uren*, George van Schoor**

*School of Electrical, Electronic and Computer Engineering, North-West University, Potchefstroom, South Africa (Tel: +27 83 504 2642; e-mail: mcterbanche@gmail.com; kenny.uren@nwu.ac.za).
**Unit for Energy Systems, North-West University, Potchefstroom, South Africa (e-mail: george.vanschoor@nwu.ac.za)

Abstract: In this paper, optimal linear control techniques are utilised to control a radio-controlled helicopter (30 size) in the AeroSIMRC simulation environment. A grey-box time-domain system identification method is used to estimate a linear state space model that operates in hover mode. Identifying the unknown parameters in the model is highly dependent on the initial values and the input data. The model is divided into sub-systems to make estimation possible. The identified state space model shows a good measure of fit compared to the simulator's flight data. A linear quadratic controller forms the inner-loop, and an optimised PID outer-loop generates attitude commands from a given inertial position trajectory. An observer estimates the unmeasured states such as blade flapping. The controller is developed in Simulink® with a plug-in written for the AeroSIMRC flight simulator. The plug-in enables Simulink® to control the simulator through a User Datagram Protocol interface for the purpose of model and controller validation. The proposed methodology facilitates a low cost test environment for new flight control algorithms. The simulation results show that the controller keeps the helicopter stable in the presence of considerable environmental disturbances and plant uncertainties.

Keywords: System identification; Linear optimal control; LQR control method;

1. INTRODUCTION

The past decade has seen a dramatic increase in the use of unmanned aerial vehicles (UAVs) (Office of the Secretary of Defence, 2011), which led to a corresponding increase in research on autonomous vehicles. In this regard, rotary winged UAVs (RWUAV) offer an especially appealing platform for researchers and developers. Their vertical take-off and landing (VTOL) capabilities make them ideal for use in urban environments and research labs. However, helicopters pose a unique challenge for control engineers. These systems are difficult to model and control since they are multiple-input and multiple-output (MIMO), are under actuated, open-loop unstable, and significantly coupled, and exhibit complex non-linear dynamics.

Mettler, Tischler, & Kanade (2001) addressed some of these modelling challenges by way of system identification. Their Comprehensive Identification from FrEquency Responses (CIFER®) model is used for helicopter control design in numerous studies. One of the key advantages of using system identification is that the experimental nature helps the engineer to better identify the dominant characteristics of the helicopter (Mettler et al., 2001). With regard to control, (Kendoul, 2012) states that PID control is still the most commonly used control technique in RWUAVs. According to (Raptis & Valavanis, 2010), non-linear and adaptive control techniques have demonstrated superior performance in controlling nonl-inear systems such as helicopters. However, (Sanahuja, Castillo, Garcia, & Lozano, 2007) states that non-linear controllers are difficult to implement and maintain in real applications. In (Valavanis, 2007), Castillo presents a linear quadratic controller for helicopter control. The simplicity of this controller facilitates field application and provides improved performance with regard to PID control. In (Ribeiro & Oliveira, 2010), a test environment is developed for the purpose of autopilot design. The controllers are developed in Matlab/Simulink®, and X-plane is used to model the aircraft flight dynamics. A stationary model aircraft is connected to Matlab® via a microcontroller and serial interface. This setup enables students and control engineers to develop flight controllers in Matlab®, test the controllers in a simulation environment, and observe the resulting hardware response. However, the professional version of this software can be costly.

This paper proposes a methodology to develop a low cost test environment for RWUAV flight controllers. This enables researchers and developers that do not have the required hardware and infrastructure with suitable low cost tools to perform model validation. In addition, this paper also focuses on modelling the simulator helicopter and designing an optimal controller for trajectory tracking. To accomplish these tasks, the paper is outlined as follows. Section 2 provides an outline of the proposed methodology. Next, Section 3 presents an overview of the system identification process and the derived state space model. Following this, Section 4 presents the development of the flight controller. In Section 5, the controller is validated in the AeroSIMRC simulation environment. Some concluding remarks and final thoughts are presented in Section 6.

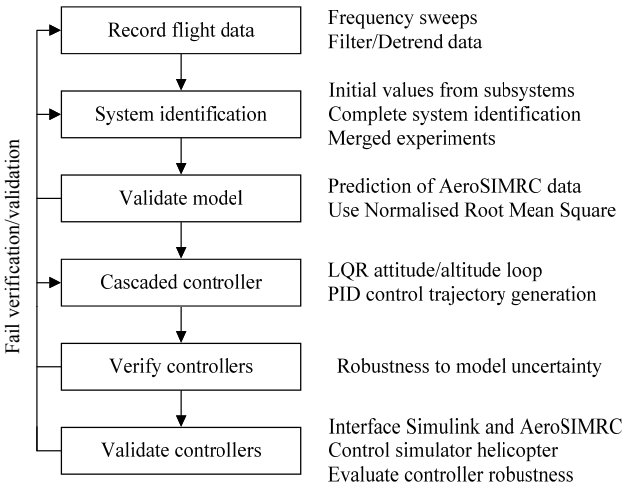

Fig. 1. Flight controller development and test methodology.

2. METHODOLOGY

Fig. 1 shows the proposed methodology to validate a helicopter flight controller in a low cost test environment. The success of each individual sub-step is determined by the quality of the flight test data. A pilot performs the flight tests manually. The data is then used to identify unknown parameters in a linear state space model. To make the identified model more general and to avoid over training on a single dataset, several experiments are combined for estimation. During the system identification process, model validation is continually performed, and only moves on when the difference between the model's performance and flight data is satisfactory. The identified model is then used to develop the inner loop of a cascaded controller. Optimal linear quadratic control theory is implemented to derive the gains for this full state feedback controller. The outer loop consists of four optimised PID controllers that generate attitude and altitude commands from a given inertial trajectory. Information regarding the variance of the identified parameters is used to perturb the model from its nominal values, and test the controller performance for modelling uncertainty. Once the controllers show satisfactory performance, the system is validated in AeroSIMRC.

3. SYSTEM MODELLING

A parameterised state space model is identified using system identification and is based on the structure developed by (Mettler et al., 2001). The hybrid nature of the model takes into account the forces created by the main and tail rotors due to a control input, and also the dynamic coupling between the rotor and fuselage. The model implemented in this paper contains only the important parameters identified by Mettler et al. (2001). To facilitate easier estimation, (Shim, 2000) divides this complex model up into simpler sub-systems.

3.1 Parameterised state space model

The model parameters are stability and control derivatives. Stability derivatives are obtained by linearising the 6 degree-of-freedom (DOF) helicopter equations of motion using a Taylor series expansion and small disturbance theory. These derivatives are commonly used in the aerospace community to evaluate the flight characteristics of aircraft. Stability derivatives give an indication of the change in force and moment acting on the aircraft due to a change in flight parameter, such as wind speed or blade flapping. Control derivatives relate the force and moment acting on the helicopter to the control surface deflection caused by inputs. The state space model is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{1}$$

with state vector

$$\mathbf{x} = [u \quad v \quad p \quad q \quad \Phi \quad \Theta \quad a \quad b \quad w \quad r \quad r_{fb}]^T, \tag{2}$$

and input vector

$$\mathbf{u} = [\delta_{lat} \quad \delta_{lon} \quad \delta_{col} \quad \delta_{ped}]^T. \tag{3}$$

The state vector describes the helicopter in the body reference frame. Fig. 2 show the variable definitions used in the model. The stability derivative matrix is

$$\mathbf{A} = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & -g & X_a & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & g & 0 & 0 & Y_b & 0 & 0 & 0 \\ L_u & L_v & 0 & 0 & 0 & 0 & L_a & L_b & 0 & 0 & 0 \\ M_u & M_v & 0 & 0 & 0 & 0 & M_a & M_b & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & \dfrac{-1}{\tau_a} & A_b & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & B_a & \dfrac{-1}{\tau_a} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Z_a & Z_b & Z_w & Z_r & 0 \\ 0 & 0 & N_p & 0 & 0 & 0 & 0 & 0 & N_w & N_r & N_{rf} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_r & K_{rf} \end{bmatrix}, \tag{4}$$

and the control derivatives matrix is

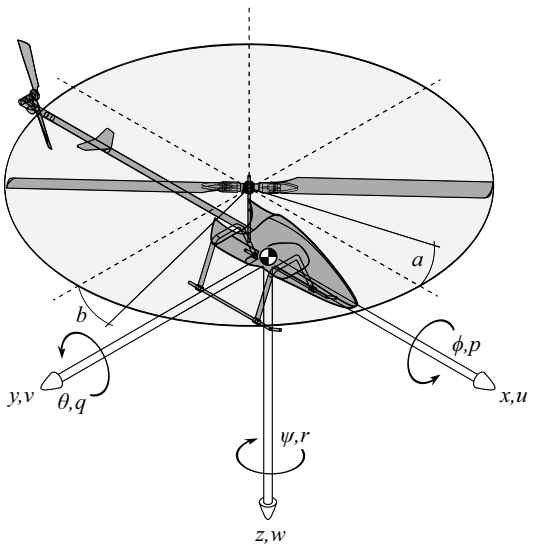

Fig. 2. State vector definitions in the body reference frame.

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ A_{lon} & A_{lat} & 0 & 0 \\ B_{lon} & B_{lat} & 0 & 0 \\ 0 & 0 & Z_{col} & 0 \\ 0 & 0 & N_{col} & N_{ped} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

3.2 System identification

Several techniques are used in the literature such as NASA's SIDPAC (Yuan & Katupitiya, 2011), CIFER® (Mettler et al., 2001), and the Prediction-Error Minimisation (PEM) for parameter estimation (Shim, 2000). However, CIFER® is an expensive program that is typically not a viable solution for low cost development. Matlab®, on the other hand, is a common development tool in the modelling and control community, and therefore, its system identification toolbox was considered the ideal choice. The parameter estimation algorithm, PEM, is considered since it gives optimal solutions; however, it is subject to longer computational times. In this case, computation time is not the limiting factor, and hence, the PEM algorithm can be implemented to enable improved results. However, the large number of unknown parameters in the state space model makes estimation difficult. Therefore, structured estimation is used to help solve this problem by excluding certain parameters from the estimation process. In spite of this, the estimation algorithm is extremely sensitive to initial values, and might therefore, not converge to the correct parameter values. The state space model is broken down into smaller sub-systems that represent different helicopter dynamics. The model sub-systems constitutes roll, pitch, heave, and yaw. Once these parameters have been estimated, the coupled systems are estimated. The roll-pitch and heave-yaw coupled systems then provide the initial values for the complete state space estimation. Dividing the system into multiple sub-systems reduces the computational complexity of the model from estimating 30 parameters simultaneously to only computing five parameters. The estimated parameters for each sub-system are used as initial values in the coupled system.

A similar identification procedure is followed for each sub-system. Flight data are recorded from AeroSIMRC and then loaded into Matlab®. The data are filtered and linear trends are removed. This excludes the trim values on the inputs and attitude angles. If the correlation between the estimated model and validation data is good, the identification procedure proceeds to the next sub-system. If the fit is poor, the flight is repeated or the model is adapted. The accuracy of the model is given by the normalised root mean square error

$$NRMSE = 100\left(1 - \frac{\|y - \hat{y}\|}{\|y - mean(y)\|}\right) \quad (6)$$

with $\hat{y}$ the model output and y the estimation data.

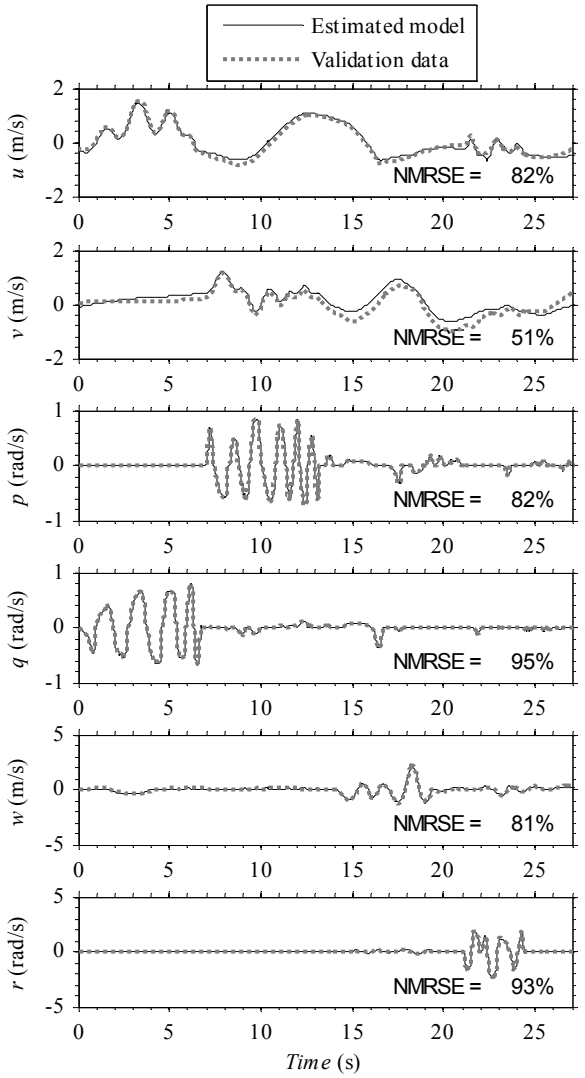

Fig. 3. Identified model fit to validation data.

Evaluation of the model without a controller is difficult since it is open loop unstable. Therefore, the estimated model is validated with a dataset that is not used in the estimation process. The model's capability to predict the system's output response for 15 steps in advance is employed to determine its validity.

3.3 Flight experiments

Each sub-system is identified by exciting only the input channels that affect that sub-system. In this regard, frequency sweeps are ideal to facilitate system identification. A chirp signal with a small amplitude at low frequencies is used to prevent the helicopter from moving out of hover. A typical set of inputs, as seen in Fig. 3, is used for model validation. As each channel is excited, the pilot uses the other inputs to keep the helicopter stable and prevent large translational velocities. This process can be automated if a flight controller is available (Sguanci et al., 2012). However, it is possible to select a set of parameters that closely match the validation data, but is not a good representation of the actual helicopter dynamics. This problem is minimised by using several sets of flight data. The data sets contain not only frequency sweeps, but also other flight manoeuvres such as figure 8's. The

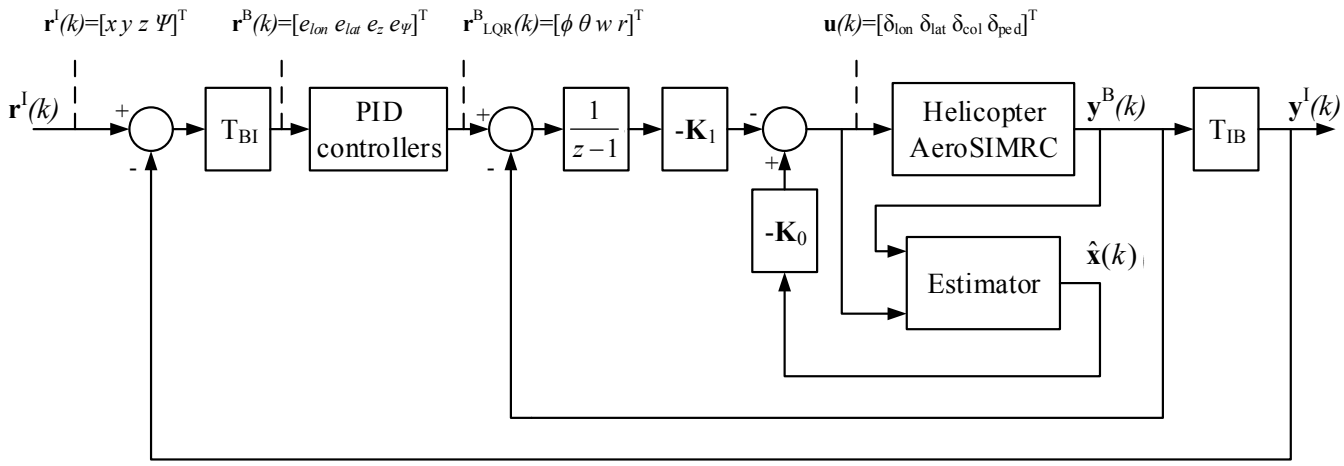

Fig. 4. Controller overview.

estimation algorithm uses the parameter variances in each of the experiments to form a weighted mean of all the models. This facilitates experimental estimates where there is the least amount of uncertainty regarding the parameter's true values.

Fig. 3 shows that the correlation between the model estimates and the helicopter response is good. The frequency sweeps employed are obtained from AeroSIMRC. On average, translational velocity predictions perform worse than angular rates. When only aerodynamic forces dictate the yaw angle, the model cannot make accurate predictions. In addition, model performance deteriorates for data sets out of hover. The accuracy of the model estimates can be improved by taking into account non-linear effects such as servo rate limits, complex inflow dynamics, or the ground effect.

4. FLIGHT CONTROLLER

The helicopter's translational motion is much slower than its angular motion, which makes it possible to use a cascaded controller with the slower outer loop controlling position and the faster inner loop for attitude control (Arain & Pota, 2011). This configuration is shown in Fig. 4 and is based on the control methodology presented by (Valavanis, 2007). The *x*, *y*, *z*, and ψ reference trajectory for the outer loop would typically be provided by a guidance controller. The outer loop then provides a roll, pitch, heave velocity, and yaw rate (ϕ, θ, *w*, *r*) reference to the inner loop. A full state feedback optimal control law is designed for the inner loop that decouples the helicopter dynamics, thereby making it possible to control the translational motion with four decoupled PID controllers.

4.1 Attitude control

The linear-quadratic regulator (LQR) algorithm provides a straightforward way to implement a MIMO controller, given that a model of the system is available. Stability is guaranteed with LQR if modelling errors and disturbances are within bounds. An advantage of using LQR for research purposes is that it can be expanded to a gain-scheduling controller and an adaptive controller (van Schalkwyk, 2008). First, the continuous time model is converted to discrete time using a zero-order hold with a sampling rate of 50 Hz. The state space model is augmented with error states to enable tracking of reference inputs (Franklin et al., 2011). The state vector is augmented with the integral of the error $\mathbf{x}_I$, for which the derivative is the tracking error **e**

$$\begin{aligned}\mathbf{x}_I(k+1) &= \mathbf{e}(k) = \mathbf{y}(k) - \mathbf{r}(k),\\ \mathbf{x}_I(k+1) &= \sum_{k=0}^{N} \mathbf{e}(k).\end{aligned} \tag{7}$$

The augmented system is

$$\begin{bmatrix}\mathbf{x}(k+1)\\ \mathbf{x}_I(k+1)\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{C} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{x}_I(k)\end{bmatrix} + \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}\mathbf{u} - \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}\mathbf{r}. \tag{8}$$

The control law is of the form

$$\begin{aligned}\mathbf{u}(k) &= -\begin{bmatrix}\mathbf{K}_a\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{x}_I(k)\end{bmatrix},\\ \mathbf{K}_a &= \begin{bmatrix}\mathbf{K}_0 & \mathbf{K}_1\end{bmatrix}.\end{aligned} \tag{9}$$

To simplify the notation, the augmented system is described by

$$\mathbf{x}_a(k+1) = \mathbf{A}_a\mathbf{x}_a(k) + \mathbf{B}_a\mathbf{u}_a(k), \tag{10}$$

with the control law

$$\mathbf{u}_a(k) = -\mathbf{K}_a\mathbf{x}_a(k). \tag{11}$$

The quadratic cost function is computed by

$$\mathbf{J} = \sum_{k=0}^{\infty}(\mathbf{x}_a(k)^T\mathbf{Q}\mathbf{x}_a(k) + \mathbf{u}_a(k)^T\mathbf{R}\mathbf{u}_a(k)). \tag{12}$$

The relative values of **Q** and **R** determine the amount of control effort that can be used to keep the states at zero. The optimal steady state gain $\mathbf{K}_a$ can be calculated recursively by solving the algebraic Ricatti equation until it converges. The performance of the controller is now dependant on the choice of the cost function. Bryson's rule can be used as a first iteration to get a stable system running (Bryson, 1975). The two weighting matrices $\mathbf{Q}_{ii}$ and $\mathbf{R}_{ii}$ can be determined with

$$\mathbf{Q}_{ii} = \frac{1}{\max(\mathbf{x}_i^2)}, \qquad \mathbf{R}_{ii} = \frac{1}{\max(\mathbf{u}_i^2)}. \tag{13}$$

The cost function weights are now dimensionless, and only the relative values of the matrices will alter the controller's response. The cost function can now be changed iteratively to reach the design criteria. Note that it is important to ensure that the actuator physical limits and rate limits are not reached since the LQR algorithm does not explicitly account for this. In addition, an estimator or observer is required to estimate the non-measured states such as rotor flapping. With the estimator added to the system, the complete inner loop is given by

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_a(k+1) \\ \mathbf{e}(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_a - \mathbf{B}_a\mathbf{K}_a & \mathbf{B}_a\mathbf{K}_a \\ \mathbf{0} & \mathbf{A}-\mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{e}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{0} \end{bmatrix} \mathbf{r}(k), \\ \mathbf{y}(k+1) &= \begin{bmatrix} \mathbf{C}_a & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{e}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \end{bmatrix} \mathbf{r}(k). \end{aligned} \tag{14}$$

4.2 Trajectory generation

A PID controller converts the x, y, z, and ψ errors in the body reference frame to the appropriate roll, pitch, heave velocity, and yaw rates. Reference trajectories are given in the inertial reference frame, however, the position errors are calculated in the body reference frame. This ensures that the error for x decreases by changing the pitch, no matter which way the helicopter is facing. The PID controllers are tuned using a simplex search method from the Matlab® optimisation toolbox. The integral time absolute error (ITAE) is minimised as the objective function. The controllers are tuned separately until each objective function's terminating condition is reached. The SISO approach followed in the outer loop is possible since the LQR controller has decoupled the helicopter attitude control.

Fig. 5 shows a 3D figure 8 trajectory with wind disturbances in AeroSIMRC. The figure shows that the controller keeps the helicopter stable in winds of up to 8 m/s. This demonstrates that the controller can perform well even for out-of-design operating conditions.

5. TEST ENVIRONMENT

To test the proposed methodology as well as the flight controller, a third party simulation package can serve as a testing platform. Game simulators are mostly used by enthusiasts to practice flying, whereas engineering flight simulators provide engineers with suitable tools to test new aircraft systems. Due to budgetary constraints, a game simulator is used which facilitates a low cost test environment. Three game simulators are commonly found in literature, Microsoft Flight Simulator, FlightGear, and X-plane. These three simulators provide external interfaces for software-in-the-loop (SITL) and hardware-in-the-loop (HITL) simulations. However, they are designed for full-size aircraft simulations and not radio-controlled (RC) aircraft. In contrast, RC helicopter simulators used by hobbyists such as Phoenix and AeroFly do not have interface capabilities. In this regard, AeroSIMRC provides interface capabilities and is specifically created for training RC pilots.

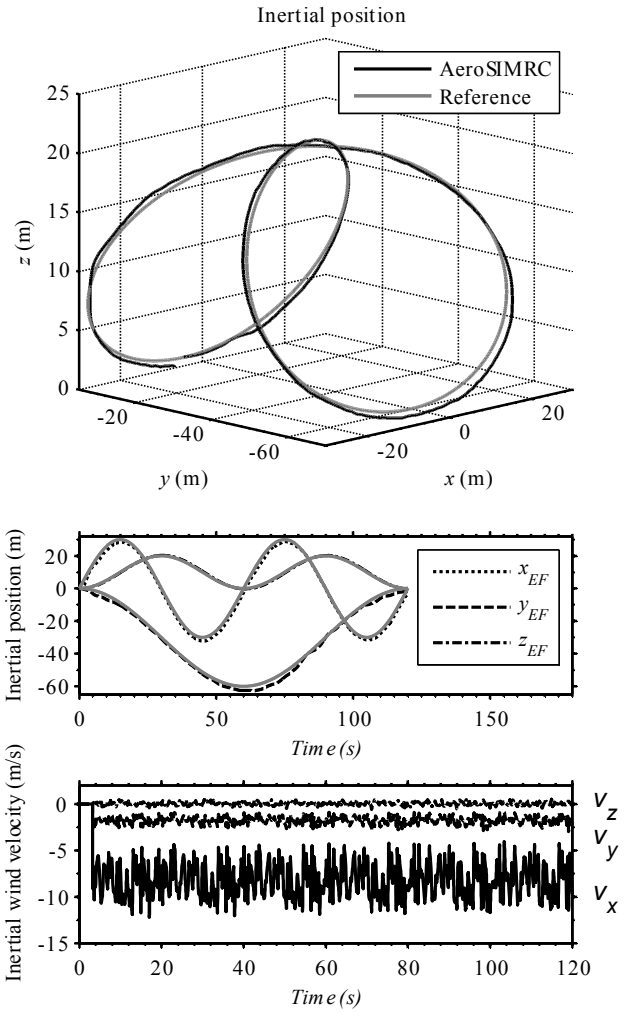

Fig. 5. 3D figure 8 trajectory with wind disturbances in AeroSIMRC (Grey: Reference. Black: Flight data).

The simulation methodology is depicted in Fig. 6. Simulink® and Matlab® have built in functions to handle serial, UDP, and TCP communication. User Datagram Protocol (UDP) is a low overhead, error-correction less transmission protocol that is ideal for real time operation and communication between applications. UDP is used to transmit data between the plug-in and Simulink®. The interface plug-in is called on every AeroSIMRC program cycle. AeroSIMRC then packages helicopter sensor measurements and sends it to Simulink® via the plug-in. The plug-in creates a binary stream of data, which is then sent to Simulink® through UDP. The data is unpacked in Simulink® and the helicopter states can be estimated. The new control inputs are calculated using the estimated states. The helicopter inputs are packaged and sent back to AeroSIMRC via the plug-in using UDP. The plug-in receives the binary stream, unpacks it into the appropriate structure, and passes it to AeroSIMRC. AeroSIMRC does not continue operation until it has received new control commands from Simulink®. The disadvantage in using game based simulators is that they were not designed for control system development. AeroSIMRC always attempts to maintain real time operation. If it cannot internally calculate the model dynamics fast enough, it will slow down the

simulation rate. This causes a change in the simulated time between function calls to the plug-in, which changes the sampling rate perceived in Simulink®. This can cause the system to become unstable. Therefore, to enforce a specific simulation rate, an in-game video is recorded during simulation. The video frame rate then serves as the system sampling rate.

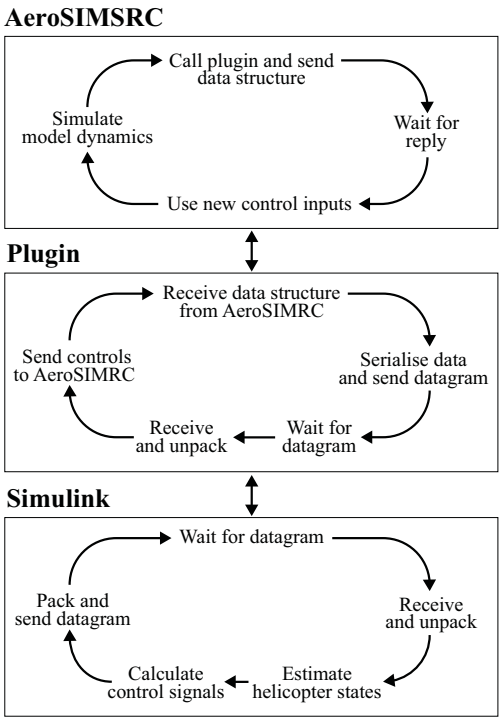

Fig. 6. Simulation flow methodology.

6. CONCLUSIONS

A simple low cost methodology is developed to test a helicopter flight controller. By using system identification, an accurate model is obtained without the need for in-depth knowledge of helicopter theory. Optimal linear quadratic control is ideal for fast attitude dynamics without being too complex for the development of an initial flight controller. An optimised PID translational controller provides the reference states needed for trajectory tracking. AeroSIMRC provided a low cost alternative to X-Plane as a platform for testing RC flight controllers.

The test environment can be upgraded to include hardware-in-the-loop capabilities. This will make it possible to develop the embedded flight controller and test hardware integration before the first real test-flight (Ng et al., 2005). In addition, the controller can be improved by employing a time-invariant LQR controller which will improve the inner loop response (van Schalkwyk, 2008). An adaptive controller capable of performing 3D manoeuvres should be the next goal. The focus should then shift to guidance and navigation (Kendoul, 2012).

ACKNOWLEDGEMENT

This work is based on the research supported in part by the National Research Foundation of South Africa (UID: 80020); The grant holder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

REFERENCES

Arain, B., & Pota, H. (2011). Flight Control of a Rotary wing UAV including Flapping Dynamics. World Congress, 18(1), 10373–10378.

Bryson, A. E. (1975). Applied Optimal Control: Optimization, Estimation and Control. Taylor & Francis.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2011). Feedback Control of Dynamic Systems. Pearson Education.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. Journal of Field Robotics, 29(2), 315–378.

Mettler, B., Tischler, M. B., & Kanade, T. (2001). System identification of a small-scale unmanned rotorcraft for flight control design. Journal of the American helicopter society.

Ng, T. L., Krishnamurthy, P., Khorrami, F., & Fujikawa, S. (2005). Autonomous Flight Control and Hardware-in-the-loop Simulator for a Small Helicopter. World Congress, 16(1), 211–216.

Office of the Secretary of Defence. (2011). Unmanned Systems Integrated Roadmap FY2011-2036.

Raptis, I. A., & Valavanis, K. P. (2010). Linear and Nonlinear Control of Small-Scale Unmanned Helicopters. (S. G. Tzafestas, Ed.)Engineering (1st ed., Vol. 45). Springer.

Ribeiro, L. R., & Oliveira, N. M. F. (2010). UAV autopilot controllers test platform using Matlab/Simulink and X-Plane. 2010 IEEE Frontiers in Education Conference (FIE), S2H–1–S2H–6.

Sanahuja, G., Castillo, P., Garcia, O., & Lozano, R. (2007). Linear and Nonlinear Control Strategies to Stabilize a VTOL Aircraft: Comparative Analysis. Intelligent Autonomous Vehicles, 6(1), 228–238.

Sguanci, M., Bergamasco, M., & Lovera, M. (2012). Continuous-Time Model Identification for Rotorcraft Dynamics. System Identification, 16(1), 816–821.

Shim, H. (2000). Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles. University of California.

Valavanis, K. P. (2007). Advances in Unmanned Aerial Vehicles. (K. P. Valavanis, Ed.) (Vol. 33). Dordrecht: Springer Netherlands.

Van Schalkwyk, C. (2008). Full State Control of a Fury X-Cell Unmanned Helicopter. Electronic Engineering. University of Stellenbosch.

Yuan, W., & Katupitiya, J. (2011). A Time-Domain Grey-Box System Identification Procedure for Scale Model Helicopters. Proceedings of the 13th Australian Conference, 1–10.