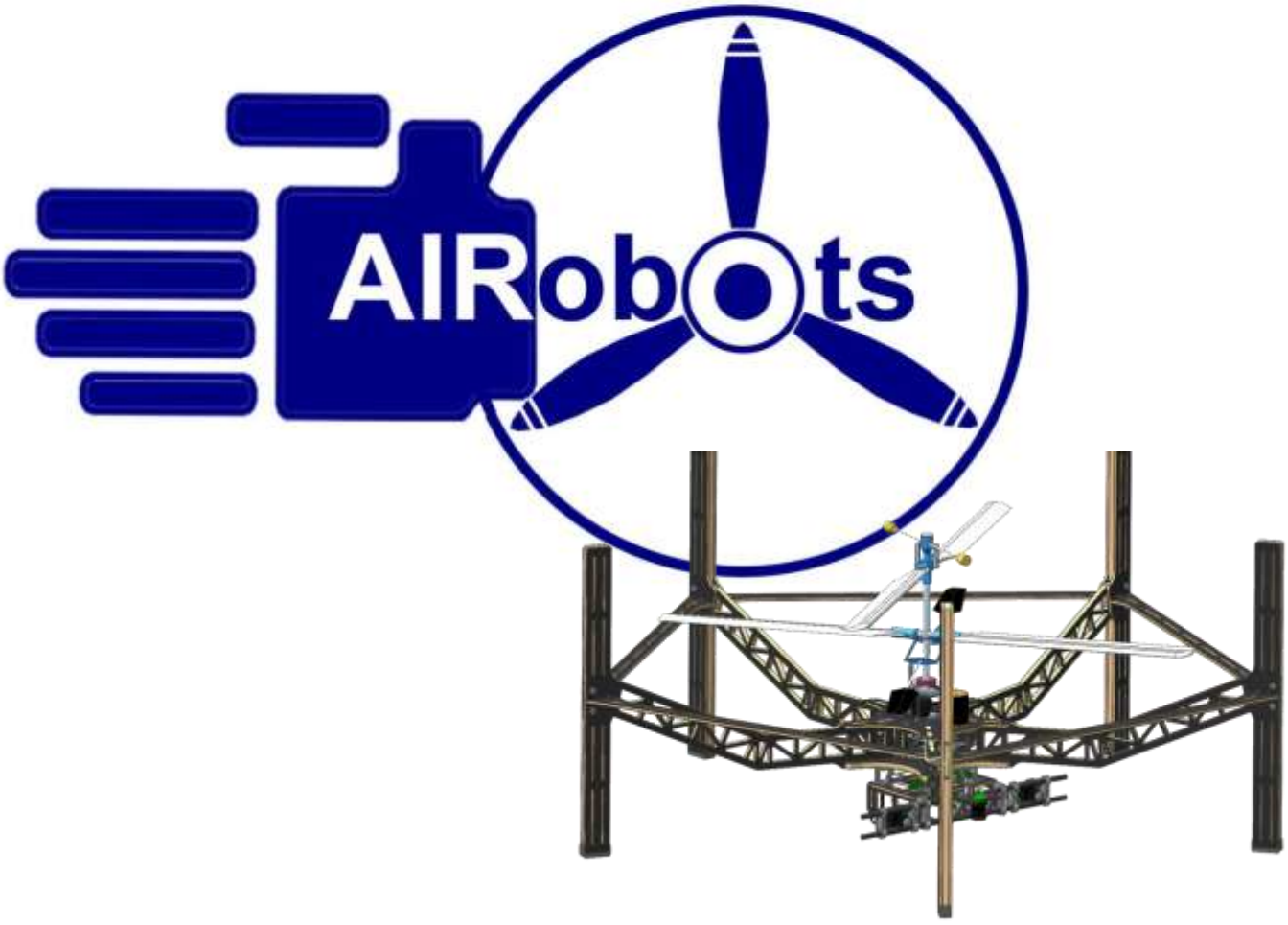
Autonomous Systems Lab

ETH Zürich

AIROBOTS Coaxial Helicopter Prototype Linear Simulator



Technical Report

AIROBOTS: Aerial Service Robotics Summer School, Zürich, 2-6 July, 2012
http://www.roboticsschool.ethz.ch/airobots
www.asl.ethz.ch

1 Introduction

An introduction to the AIROBOTS Coaxial (ACX) Prototype Linear simulator (`ACX Simulator`) is provided. This document serves as a short technical reference about the methods employed in order to derive the models of the ACX dynamicsand the controllers utilized as well as a brief introduction to the simulator itself from an user point–of–view. The models were derived using frequency–domain system identification techniques inspired from the experience of the aerospace community about the proper methods to conduct the identification experiments, how to prepare the data, which methods to use during the optimization step and how to validate the identified model. A simplified version of the main control approach utilized in practice is explained followed from some short description about alternative controllers implemented within the `ACX Simulator`. Finally, the MATLAB®–Simulink® simulator blocks are explained and an overview of the provided post–analysis and parametric/nonparametric identification functions is given.

2 ACX Simulator Coupled Rotors/Flybar/Fuselage Identified models

The AIROBOTS Coaxial prototype is a complex system with coupled nonlinear rigid body/rotors (lower,upper and Flybar) dynamics. Despite its nonlinear nature a high–order linearized hovering approach is found to be accurate enough for a sufficient part of the industrial inspection–related mission envelope of the ACX. Moreover, the huge complexity of the nonlinear model makes it unsuitable for a lot of tasks including control design and uncertainty analysis. In order to acquire a more convenient model for analysis and control design, a linear model is derived. Typically, in most UAV applications, the quasi-steady 6-DOF model is utilized [1]. Despite its attractive simplicity, this model is not valid for the ACX due to its rotors configuration. For hingeless rotors with long diameter and relatively flexible blades, the rotor and fuselage modes become highly coupled, and thus a model structure that also accounts for the flapping dynamics must be derived [2]. The approach of the coupled rotor/fuselage dynamics is called the *hybrid model*.

In the pitch and roll degrees of freedom, the hybrid model combines a physical model of the coupled fuselage/regressive flap dynamics of the lower and higher rotors as well as the flybar, with a quasi-steady stability derivatives model for accurate mid–high and low frequency dynamics modeling. The inclusion of explicit flapping dynamics replaces the conventional quasi–steady rotor derivatives associated with the vehicle's angular motion and the control inputs. This hybrid formulation has the advantage of being more accurate than the quasi–steady model in a wider frequency area, while also being physically consistent. The yaw–heave dynamics are modeled using a quasi–steady model since the coning modes are comparatively negligible [3]. The hybrid model is derived through linearization of the nonlinear dynamics around the hovering operating point and takes the following form for the attitude and heave motions:

$$
\begin{aligned}
\dot{\mathbf{x}} &= \left[\begin{array}{c|c} \mathbf{A}_{\psi,\mathbf{z}} & \mathbf{0}_{\mathbf{10}\times\mathbf{10}} \\ \hline \mathbf{0}_{\mathbf{4}\times\mathbf{4}} & \mathbf{A}_{\phi,\theta} \end{array}\right] \mathbf{x} + \left[\begin{array}{c} \mathbf{B}_{\psi,\mathbf{z}} \\ \hline \mathbf{B}_{\phi,\theta} \end{array}\right] \mathbf{u} \\
\mathbf{x} &= [z, w, \psi, r, \phi, p, \theta, q, \alpha, \beta, \gamma, \delta, \varepsilon, \zeta]^T \\
\mathbf{u} &= [\phi_{1c}, \phi_{1s}, h_{sp}, \Omega_{mot}]^T
\end{aligned}
\tag{1}
$$

$$\mathbf{A}_{\psi,\mathbf{z}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -w_w & 0 & -w_r \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -r_r \end{bmatrix},$$

$$\mathbf{A}_{\phi,\theta} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -p_\beta & 0 & -p_\delta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_\alpha & 0 & q_\gamma & 0 & 0 & 0 \\ 0 & \alpha_p & 0 & -\alpha_q & -\alpha_\alpha & \alpha_\beta & 0 & 0 & -\alpha_\varepsilon & -\alpha_\zeta \\ 0 & \beta_p & 0 & \beta_q & -\beta_\alpha & -\beta_\beta & 0 & 0 & \beta_\varepsilon & -\beta_\zeta \\ 0 & -\gamma_p & 0 & -\gamma_q & 0 & 0 & -\gamma_\gamma & -\gamma_\delta & 0 & 0 \\ 0 & \delta_p & 0 & -\delta_q & 0 & 0 & \delta_\gamma & -\delta_\delta & 0 & 0 \\ 0 & 0 & 0 & -\varepsilon_q & 0 & 0 & 0 & 0 & -\varepsilon_\varepsilon & -\varepsilon_\zeta \\ 0 & \zeta_p & 0 & 0 & 0 & 0 & 0 & 0 & \zeta_\varepsilon & -\zeta_\zeta \end{bmatrix},$$

$$\mathbf{B}_{\psi,\mathbf{z}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -w_{h_{sp}} & -w_{\Omega_{mot}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -r_{h_{sp}} & -r_{\Omega_{mot}} \end{bmatrix},$$

$$\mathbf{B}_{\phi,\theta} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -p_{h_{sp}} & -p_{\Omega_{mot}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_{h_{sp}} & q_{\Omega_{mot}} \\ -\alpha_{\phi_{1c}} & -\alpha_{\phi_{1s}} & 0 & 0 \\ \beta_{\phi_{1c}} & -\beta_{\phi_{1s}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\varepsilon_{\phi_{1c}} & -\varepsilon_{\phi_{1s}} & 0 & 0 \\ \zeta_{\phi_{1c}} & -\zeta_{\phi_{1s}} & 0 & 0 \end{bmatrix}$$

For this model the states $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta$ correspond to the longitudinal and lateral blade flapping of the lower, upper rotor and the flybar correspondingly and are not measurable but they are detectable, the input Ω_{mot} is measured with an optical speed encoder while the inputs $\phi_{1c}, \phi_{1s}, h_{sp}$ are the values produced from the open or closed loop commands.

The model given in equation (1) contains 49 unknown parameters, with the majority of them being very complex to compute from pure physics. System identification based only on flight data is the natural tool to overcome this difficulty. In order to tackle this problem a Frequency–Domain approach was selected. Frequency–domain system identification poses significant advantages compared to time domain identification, including: a) unbiased frequency–response estimates when data contain process and output measurement noise, a fact that drops the requirement to identify a noise model even when measurement noise is close to the vehicle's structural vibrations, b) use of the coherence function as an unbiased metric of nonparametric accuracy, level of excitation and linearity of the system response, c) selection of different frequency ranges for different inputs or outputs which is helpful due to the frequency seperation of the roll/pitch and yaw/heave dynamics, d) easier identification of systems with unstable dynamics. These advantages are very important especially for rotorcraft identification [2].

Frequency–Domain SYStem IDentification (FD–SYSID) is a complicated optimization process that also needs careful conducting of the experiments and data preparation. To make this process systematic a FD–SYSID tool was developed which highly automates the whole process. A Flow diagram of the ACX FD–SYSID tool is presented in Figure 1.

From the state–space matrix representation of the linear system it is shown that the pitch/roll and yaw/heave dynamics are decoupled. Consequently two sets of experiments were conducted. In the first set the pitch/roll rates are excited using open–loop automated frequency–sweeps (chirp signals) in all the desired frequency area (0.1–12 rad/s) while yaw and heave are closed–loop controlled. Similarly, in the second set of experiments, yaw and heave are excited using open–loop automated frequency sweeps in the area of 0.05–6 rad/s, while roll and pitch are automatically controlled. The time duration of the experiments is long enough so that the slowest dynamics are captured and if needed different experiments are concatenated, a method valid as long as all experiments start and end at trim [2]. Aerospace engineering experience [2] indicates that the required flight duration T_{rec} is:

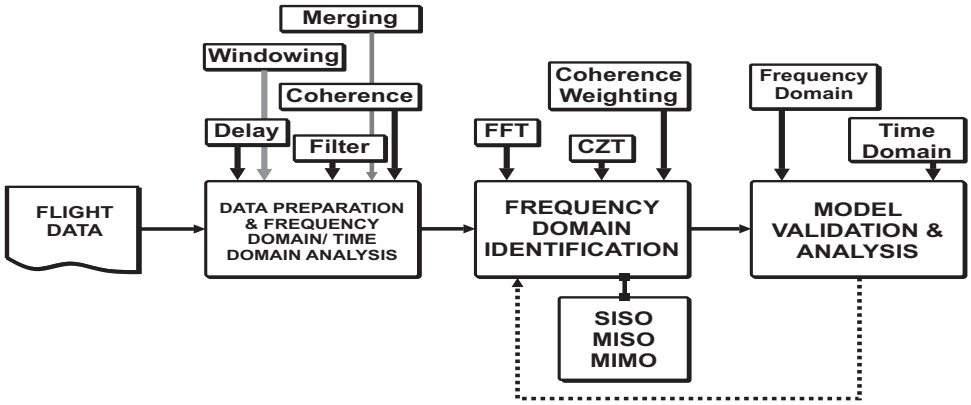


Figure 1: ACX System Identification Tool Flow Diagram

$$T_{rec} \geq (4 \text{ to } 5) T_{\max} \tag{2}$$

,where $T_{\max}$ is the slowest expected oscillation of the specific degree of freedom. The quality of the recorded data is analyzed using the magnitude squared coherence $\gamma^2_{u_i y_j}$ between all the inputs u_i that excite a specific output y_j and this output. It is known [2, 3] that if

$$\gamma^2_{u_i y_j} \geq 0.6 \tag{3}$$

within the selected–desired frequency area, then a sufficient and proper excitation is achieved and the response can be modeled as a linear system since there is frequency–matching. Within the framework of the ACX FD–SYSID tool each experiment was analyzed and was used for the identification algorithm only when this criterion is satisfied. Additionally the method of overlapped windowing [4] is used in order to reduce the level of random error in the spectral estimates by averaging the rough estimates for multiple segments of data. Within the ACX FD–SYSID tool hanning [4] windows are utilized, while the following rules of thumb are adopted from the aerospace industry experience [2]:

$$T_{win}^{nominal} = 2T_{\max}, \;\; T_{win}^{\max} = 0.2T_F, \;\; T_{win}^{\min} = 20\frac{2\pi}{\omega_{\max}} \tag{4}$$

,where $T_{win}^{nominal}$ is the nominal window size, $T_{win}^{\max}$, $T_{win}^{\min}$ are the maximum and minimum window lengths respectively, T_F is the merged experiments length and $\omega_{\max}$ is the maximum frequency of interest for the specific output.

Once the stage of flight data recording and data preparation is completed, the frequency response of the data is computed using the Fast Fourier Transform (FFT) or the Chirp–Z transform (CZT) [4], an alternative way to compute the frequency response that gives the capability to zoom at specific frequencies, a fact that provides the potentiality to focus mainly on the helicopter oscillations without being influenced from the noise or non–dominant nonlinearities. The ACX FD–SYSID tool has pre-programmed all the required structure for Single–Input/Single–Output, Multi–Input/Single–Output and Multi–Input/Multi–Output (MIMO) state–space grey–box identification. The first two structures act as a first step to identify an initial model of the dominant effects or analyze the influence of each input to the outputs. However, the ultimate goal especially for the ACX is MIMO identification. The solution of the MIMO identification problem involves determining the model matrices $\mathbf{A}_{\phi,\theta}, \mathbf{A}_{\psi,z}$, $\mathbf{B}_{\phi,\theta}, \mathbf{B}_{\psi,z}$ and the input/actuation delays τ that produce a frequency–response matrix $\hat{\mathbf{T}}_c$ that most closely matches the frequency responses $\mathbf{T}$ obtained from the experimental results. The novelty compared to the classical methods is that the coherence and the CZT provide the capability to weight the identification optimization algorithm in a way that the subset of the frequency response that is highly excited from the input plays a more important role. This prevents errors in the identification due to overfocusing in nonlinearities outside the helicopter useful flight envelope or off–axis responses. The associated cost function to be minimized takes the following form [2]:

$$\mathbf{J} = \sum_{l=1}^{n_{TF}} \sum_{\omega_1}^{\omega_{n_\omega}} W_\gamma(\omega_i)[W_g(|\hat{\mathbf{T}}_c(\omega_i)| - |\mathbf{T}(\omega_i)|)^2 + W_p(\angle\hat{\mathbf{T}}(\omega_i)_c - \angle\mathbf{T}(\omega_i))^2],$$

$$W_\gamma(\omega) = [1.58(1-\exp^{-\gamma^2_{u_i,y_j}})]^2, \;\; W_g = 1.0, \;\; W_p = 0.01745, \tag{5}$$

where the function $W_\gamma(\omega_i)$ is the weighting function that depends on the coherence between input and output at frequency ω_i, n_ω is the number of the frequency points, ω_1 and ω_{n_ω} are the starting and ending frequencies of the fit and n_{TF} is the number of the transfer functions that are produced from the all the input/output relations of the MIMO system.

The ACX FD–SYSID tool is developed over MATLAB® making use of the functionality that already exists within the System Identification Toolbox and the Signal Processing Toolbox. Additional tools and algorithms were developed to incorporate the coherence metric in the identification procedure, the coherence weighting and the aforementioned frequency–domain optimization algorithm, the Chirp–Z transform and the method of overlapped windowing.

The application of the aforementioned methods in the ACX hybrid model was successful. Despite the complexity of the configuration especially in the roll–pitch degrees of freedom, where the higher–order flapping dynamics are present, the identification results (validated in datasets not used during the identification procedure) both in terms of time–domain fitting and coherence are very good as illustrated in Figure 2. Additionally, input delays were modeled in order to capture the actuation dynamics. It must be noted that what is actually identified are the body rates and velocities, while the absolute angle and position is derived through integration. The measurements were based on the VICON motion capture system operating at $T_s = 0.01$s sample period. The fitting percentage is defined as $Fit = 100(1-|y-\hat{y}|/|y-E(y)|)$, where y is the experimental output, $E(y)$ its mean value and $\hat{y}$ is the predicted response. During the optimization the covariance matrix is computed as a metric of the parameter certainty.

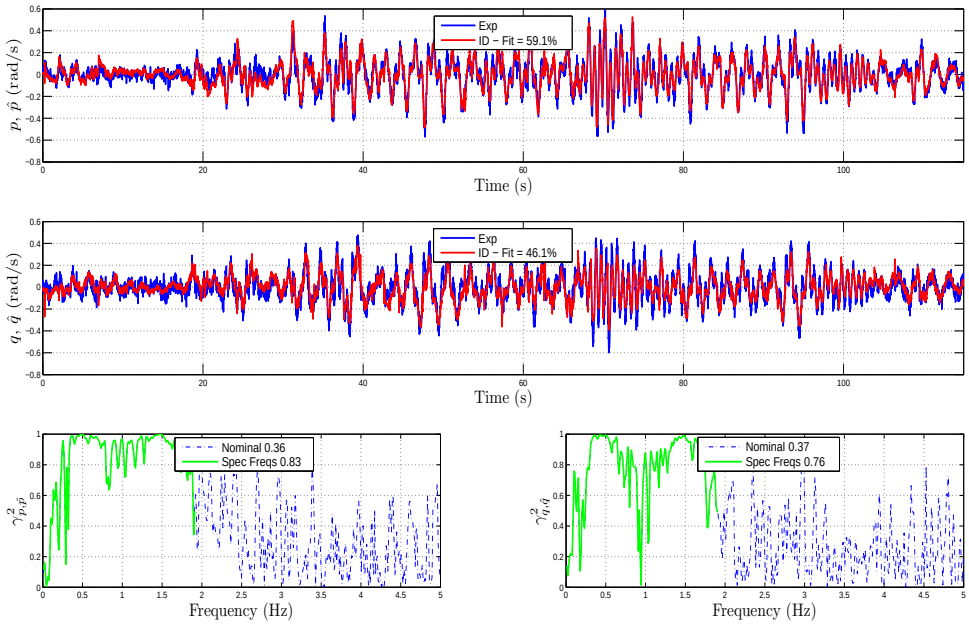


Figure 2: Time–Domain and Coherence Validation between the experimental responses of p, q rates and their estimates from the identified model $\hat{p}$, $\hat{q}$

Similarly the identification of the yaw–heave subsystem was achieved. For this subsystem the linear model is very simple keeping a first order structure since the effect of downwash was neglected. This poses some limitations in the identification accuracy, but keeps the model close to physical reality. In the literature it is common to use higher–order physically not meaningful data [5]. As a first approach it was selected to put the required effort to achieve a physically meaningful model. Other physical phenomena including the servo response quantization and

the backlash of the swashplate kinematics were modeled based on ground test–bench experiments. A validation plot indicating the sufficient accuracy despite the simplicity of the model is shown in Figure 3. Similarly to the p,q rates identification, input delays were also modeled to capture the servo dynamics.

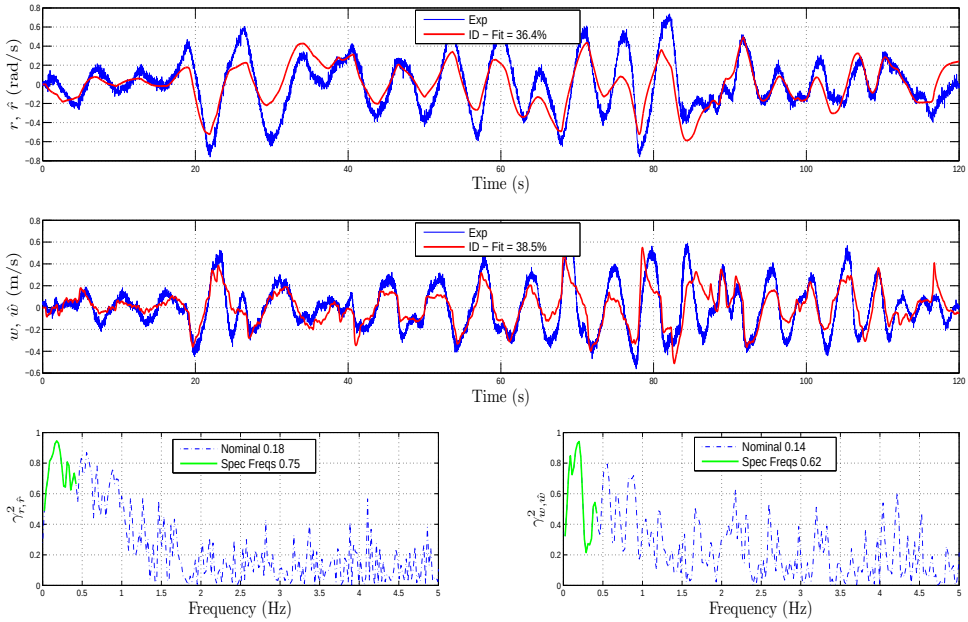


Figure 3: Time–Domain and Coherence Validation between the experimental responses of r rate, w velocity and their estimates from the identified model $\hat{r}$, $\hat{w}$

The aforementioned model for yaw and heave captures the dominant physical effects. However, there were segments of the flight response that the model only partially captured the actual response. This is mainly due to unmodeled nonlinearities and off-axis responses from the other inputs. A model with increased accuracy can be derived at the expense of going away from the physicality and increasing the system order. The new model structure is the same with the previous one for yaw–heave except that the relation between the swashplate height h_{sp} and the yaw rate r is modeled as a transfer function with 3 poles, 1 zero, and one input delay. Figure 4 shows the corresponding validation plot, where clearly higher accuracy has been achieved. This model is useful for high–bandwidth control tuning.

Based on the aforementioned results, it is clear that the ACX hybrid model was successfully identified and the model accurately captures the vehicle higher order dynamics around hover. Through the development of the ACX FD–SYSID tool the identification problem has been systematically confronted. This tool now automates the process, from data analysis and quality check, to parametric or nonparametric frequency domain system identification to model validation.

3 ACX Simulator Motion Controllers

The `ACX Simulator` implements full six–degrees of freedom control for the AIROBOTS Coaxial prototype. The main set of supported controllers are gain–scheduled PID loops with additional feedforard and on–path compensators and functionality which is related with special properties of the system such as the off–axis responses and the presence of a resonance pole inside the useful bandwidth of the roll/pitch coupled rotors/flybar/fuselage dynamics. Figure 5 indicates the main gain–scheduled control structure for the roll and pitch dynamics. It must be noted that the Flybar acts as a pseudo–rate control thus providing some sort of damping to the system. However coupled dynamics of the Flybar with the body constraint the bandwidth of the system and also lead to the presence of a resonance pole at relatively low frequencies (approximately at 1.41Hz).

The resonance pole requires the design of a special compensator in order to minimize its effects and achieve efficient and safe flight response. The solution followed so far is based on the design of a Notch filter centered at the resonance frequency which was identified nonparametrically. Figure 6 indicates the shape of the bode

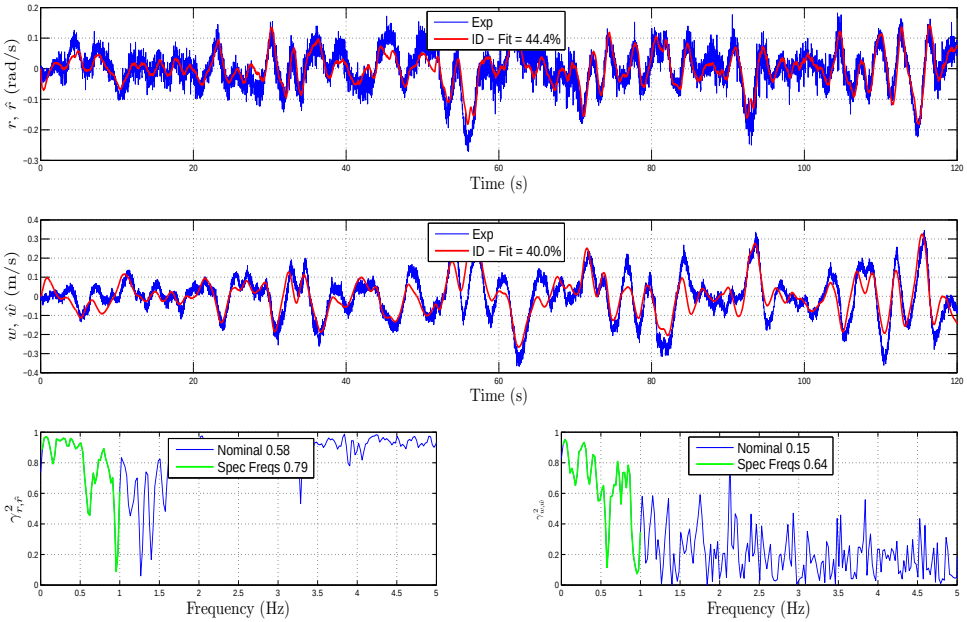


Figure 4: Time–Domain and Coherence Validation between the experimental responses of r rate, w velocity and their estimates from the higher–order identified model $\hat{r}$, $\hat{w}$

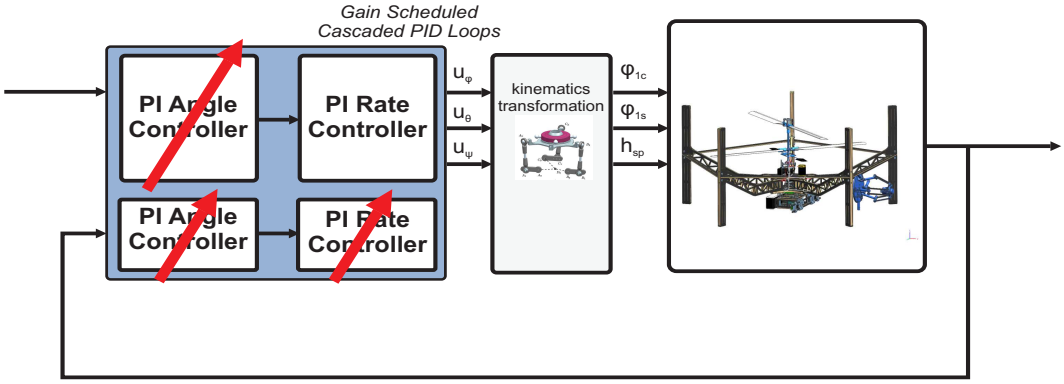


Figure 5: Block–diagram of the ACX Attitude Gain Scheduled PID control structure.

response of the notch filter and its influence in a recorded flight experiment. Once the resonance pole is effectively compensated, the derivative term of the PID loop could be tuned. It is worth noting that derivative action increases the sensitivity of the system at frequencies close to the resonance peak and thus careful tuning is required.

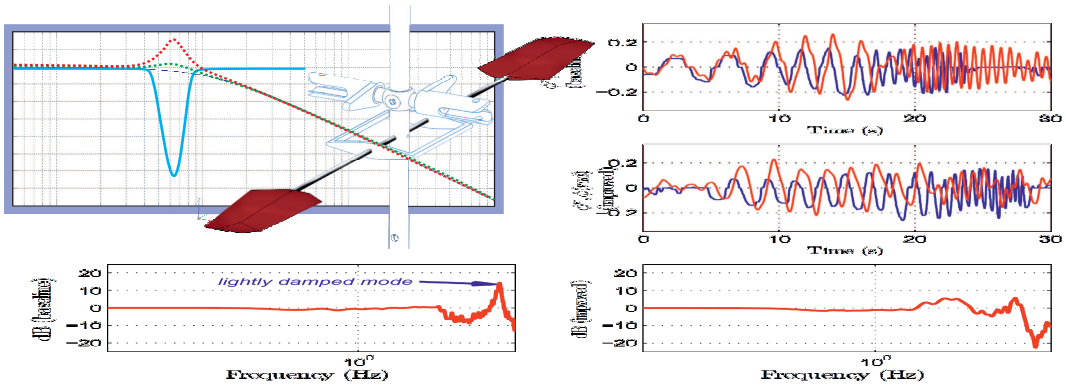


Figure 6: Resonance pole compensation.

The off–axis responses compensation consists in the implementation of feedforward terms between the normalized roll and pitch control actions. The feedforward compensator is a DC–gain cascaded with a washout–filter due to the frequency dependency of the off–axis response. A sketch of the characteristic off–axis response combined with the structure of the off–axis compensation is depicted in Figure 7.

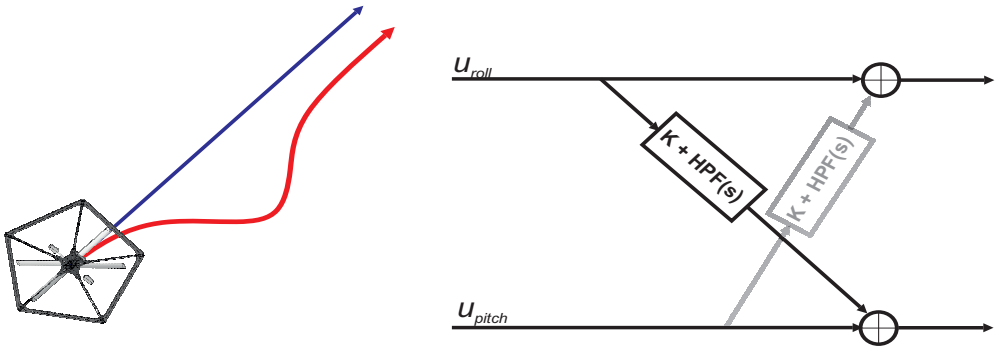


Figure 7: Off–axis characteristic response and structure of the feedforward compensators.

Once the gain–scheduled PID loops are tuned and resonance/off–axis problems are compensated , one can treat the ACX attitude dynamics as an almost linear system. This enables the capability to design the velocity/position and trajectory controllers. The position/velocity control loops consist of cascaded PID loops, which essentially means that acceleration is also used. The velocity proportional gain is also gain–scheduled. Trajectory control is achieved based on a twofold control actions which keeps constant velocity in the direction defined between two waypoints while regulating the distance of the cross–axis position to zero, as shown in Figure 8. In order to improve the motion characteristics, a polynomial approximation of the trajectory is utilized.

Finally, yaw–heave control is achieved in a gain–scheduled PID manner for both degrees of freedom. However, due to the influence of the swashplate height position to heave dynamics a feedforward compensator based on a linear predictor of this input/output relation is also utilized to compensate for this particular effect of the ACX.

The `ACX Simulator` also implements a baseline PID controller for each degree of freedom in order to compare improved control schemes with this typical approach. The gains used in the simulator are slightly higher than those tested in practice due to the partial capturing of the resonance pole and off–axis effects. Moreover, model–based controllers were also designed and are provided within the `ACX Simulator`. The model–based controllers include an LQG controller for the roll/pitch dynamics, LQI controller for the yaw dynamics and a

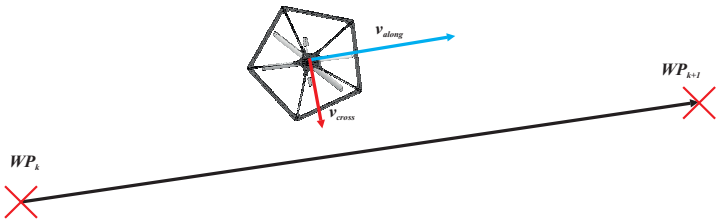


Figure 8: Trajectory constant on–axis velocity/zero cross axis position error control scheme.

PID controller optimized for increased robustness despite the induced performance degradation. The user may implement his own control laws.

4 Use of the ACX Simulator

The `ACX Simulator` is implemented using MATLAB®–Simulink® while it is possible to connect with Blender Open–Source 3D content creation suite for high quality visualization and possible environmental interaction feedback. It uses the identified state–space representations of the ACX open–loop dynamics in combination with simplified versions of the experimentally utilized control laws. Figure 9 illustrates the main Simulink® view of the simulator.

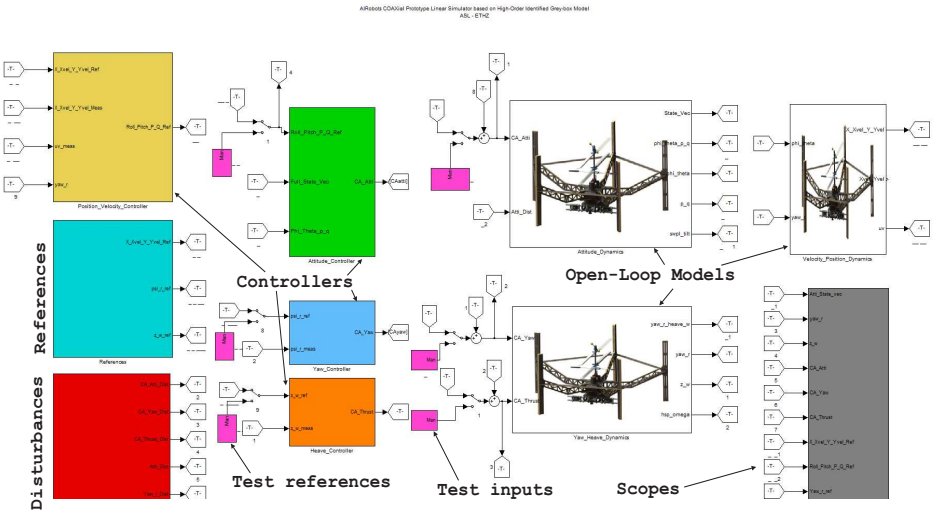


Figure 9: ACX Simulator Simulink® Model main window

To initialize and run the `ACX Simulator` execute the following steps:

1. Unzip the `ACX Simulator` files to your MATLAB folder making sure that the top directory is called `ACX_Simulator`
2. From MATLAB's enter the main directory `ACX_Simulator` and from there open the Simulink file named as `ACX_Simulator.mdl`. The `ACX_Simulator` will then automatically initialize and MATLAB's path will be updated. You may also need to add the directory `Other_Functions` at your MATLAB path.
3. In the model callbacks enable (comment–out) or disable (comment) the automated plotting function named `ACX_SIM_AutoPlotting.m`. Note that the automated plotting function requires some proper initial set–up and works only for automated references and not when the manual test–inputs are used.
4. Further documentation can be found by typing `doc FUNCTION_NAME`.

The simulator blocks have some modularity and the user can design and implement his own control laws or test the quality of models derived with alternative ways without the need to change the structure of the blocks. Automated plotting functionality is implemented in order to evaluate the performance in terms of: a) reference tracking for all outputs and coherence–based frequency matching, b) coherence and random error check between the control signals and the body rates and velocities and c) nonparametric bode estimation. The so–called `OPTIONS` objects must be properly defined so that they correspond to the characteristics of each particular degree–of–freedom and the simulation characteristics (especially the selection of the slowest possible frequency and thus the maximum period in relation with the overall simulation length). The directory tree of the `ACX Simulator` is shown in Figure 10.

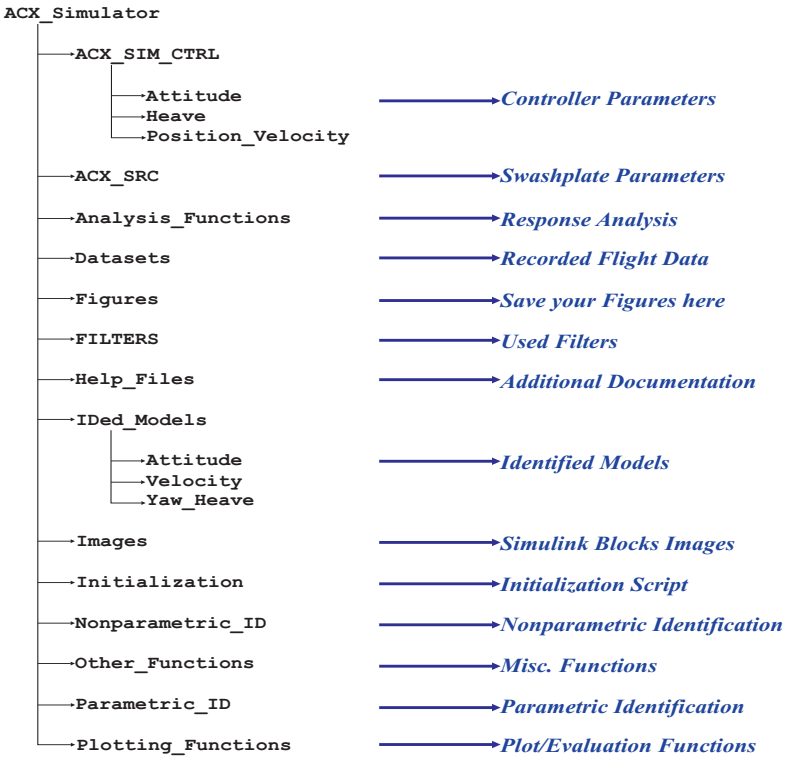


Figure 10: ACX Simulator directory tree

Some limited parametric and nonparametric identification functionality is provided based on MATLAB®'s System Identification Toolbox and additionally implemented functions. Parametric identification requires center assumptions that must be made including the following: a) what model order is necessary to capture the key dynamics, b) how highly coupled are the dynamics degrees of freedom, c) what is the proper structure of the equations of motion, d) what are good initial guesses for the identification parameters. The provided functions are based on MATLAB®'s System Identification Toolbox and aim to tackle the problems of:

- Single Input - Single Output Transfer Function Identification
- Single Input - Single Output 2nd order State–Space Model Identification
- Single Input - Single Output 3rd order with 1 zero State–Space Model Identification
- Multi Input - Single Output quasi–steady attitude identification
- Multi Input - Single Output higher order State–Space Model Identification

All of the functions can be used with time–domain and fast–fourier transform based or chirp–Z transform based frequency–domain objects. The user may benefit from these functions and the recorded flight datasets contained in the homonym directory in order to derive simpler representations of the modeled dynamics and evaluate their performance or even use them as a basis for control design purposes.

Through the utilization of this functionality and the recorded flight datasets contained in the homonym directory, the user may computed simplified parametric models and use them in the process of computing simplified model–based control laws, or estimate critical system characteristics like the resonance frequency using nonparametric identification methods.

As mentioned before, the `ACX Simulator` also contains some limited nonparametric functionality. Nonparametric identification is concerned with characterizing only the measured input–to–output behavior of the aircraft dynamics, not the nature of the aircraft equations of motion. Examples of nonparametric modeling are impulse or step responses (time–domain) and frequency responses (frequency–domain), which are both derived directly from the flight data. In either case no assumptions are required about the structure off the dynamic model. Nonparametric system–identifiction modeling provides excellent insight into the key aspects of the aircraft dynamics and can be used in order to understand the system before moving to the more complex parametric modeling state or as a control tuning aid.

Additional functionality in order to properly analyze and prepare the data for identification is also provided. Use the two scripts `ACX_SIM_ID_SISOA.m` and `ACX_SIM_ID_MISOA.m` as guidelines on how to use the main functions.

The nonparametric identification is related mostly with estimations on the input–output relation and the frequency response of the system computed based only on experimental data without the need of a particular model structure. The script named as `ACX_SIM_Nonparametric_ID.m` should be used as a guideline.

Finally, as noted before, the `ACX Simulator` can connect with Blender and the Multi Open Robots Simulator (MORSE) engine in order to provide a framework for 3D visualization, motion simulator and environmental interaction feedback. Figure 11 illustrates this feature in action:

Figure 11: Visualization of the AIROBOTS Coax and its environment using Blender

This visualization and dynamics simulation framework provides a wide set of possible advantages including: a) intuitive understanding of the aircraft flying qualities, b) means of force feedback when environmental interaction is considered, c) a way of initial remote–controlled flights training.

5 Conclusion

The main characteristics of the AIROBOTS Coaxial helicopter prototype linear simulator were introduced. Some details about the methods used in order to identify the free–flight dynamics of the vehicle were given followed by a summury of the utilized control schemes. Finally, the main technical details of the `ACX Simulator` were briefly described from an user point–of–view.

For further information about the AIROBOTS Coaxial Prototype, its design, scope and mission profile or the utilized nonlinear/linear modeling and control strategies, you may contact:

1 Christoph Huerzeler (christoph.huerzeler@mavt.ethz.ch)

2 Kostas Alexis (konstantinos.alexis@mavt.ethz.ch)

For further information about the MORSE–Blender framework you may contact:

1 Anna Chiara Bellini (annachiara.bellini@gmail.com)

We would also like to thank Daniel Grieneisen for his help in integrating the `ACX Simulator` with the MORSE–Blender framework.

References

[1] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using lagrange form dynamics," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1, pp. 127–151, 2009.

[2] Mark B. Tischler, Robert K. Remple, *Aircraft and Rotorcraft System Identification: Engineering methods with Flight-Test examples*. American Institute of Aeronautics and Astronautics (AIAA).

[3] B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft*. Kluwer Academic Press, 2003.

[4] R. G. Lyons, *Understanding Digital Signal Processing*. Prentice Hall, 2010.

[5] M. B. Gerig, "Modeling, guidance, and control of aerobatic maneuvers of an autonomous helicopter," Ph.D. dissertation, ETH Zurich, 2008.