

# Parameter Optimization and Model Validation of Quanser AERO using Modelica and RaPIId

Eric Segerstrom, Meaghan Podlaski, Abhijit Khare, and Luigi Vanfretti  
*Rensselaer Polytechnic Institute, Troy, NY, 12180*

This paper presents an overview of the process for optimizing key electrical, mechanical, and aerodynamical parameters of a multi-domain model of the Quanser AERO, a reconfigurable, benchtop, twin-rotor system using RaPIId, an open-source, modular toolbox for model validation and parameter estimation in MATLAB and Simulink. A particle swarm optimization algorithm and a constrained nonlinear optimization algorithm are compared for their efficacy in optimizing the aforementioned parameters of the AERO model in an effort to more accurately simulate the behavior of the device.

## I. Nomenclature

|              |   |                                     |
|--------------|---|-------------------------------------|
| $\epsilon$   | = | error                               |
| $\epsilon_0$ | = | error tolerance                     |
| $\tau_e$     | = | electrical torque                   |
| $\omega$     | = | motor velocity                      |
| $c_m$        | = | mechanical torque constant          |
| $E$          | = | back EMF                            |
| $F_D$        | = | drag force                          |
| $F_T$        | = | thrust force                        |
| $i_m$        | = | current through DC motor            |
| $J_r$        | = | moment of inertia of rotor          |
| $J_s$        | = | moment of inertia of stator         |
| $J_t$        | = | total moment of inertia of DC motor |
| $k_d$        | = | drag coefficient                    |
| $k_t$        | = | thrust coefficient                  |
| $L_a$        | = | armature inductance of motor        |
| $R_a$        | = | armature resistance of motor        |
| $V_m$        | = | motor voltage                       |
| $y$          | = | output                              |

## II. Introduction



Fig. 1 The Quanser AERO in a 1-degree-of-freedom configuration.

UNMANNED aerial vehicles (UAVs) are becoming increasingly ubiquitous in a variety of industrial, commercial, and agricultural settings, and as such, there is a growing opportunity to further develop these systems efficiently and effectively through modeling and simulation. To this end, representative and validated models of smaller yet similar systems can serve as a scalable and illustrative use case in defining best practices in the multi-domain modeling of UAVs. One such system is the Quanser AERO, a dual-rotor system pictured in Fig. 1. The AERO can be reconfigured for several aerospace systems including a 1-degree-of-freedom (DoF) single-input, single-output (SISO) system, a 2-DoF multiple-input, multiple-output (MIMO) system, and half-quadrotor simulations.

In this paper, key electrical, mechanical, and aerodynamical parameters of an object-oriented, multi-domain model of the Quanser AERO in its 1-DoF SISO configuration are optimized using a particle swarm optimization (PSO) algorithm and MATLAB's *fmincon* optimization algorithm. In optimizing these parameters, it is expected that the model will behave more similarly to its physical counterpart than the pre-optimized model. As such, the optimized model is then validated by comparing its output characteristics to those of the original model and the physical system. This validated model can then be used in several different applications including controller design, simulation, and further optimization.

### A. Related Works

Several studies have been conducted regarding the parameter optimization of twin-rotor and quadrotor systems using PSO algorithms. A dynamic spread factor PSO algorithm is compared with three other global PSO algorithms for their efficacy in estimating the parameters of a hovering TRMS model in [1]. The paper determines that the dynamic spread factor PSO algorithm achieved a lower mean standard error according to the objective function assigned than any of the global PSO algorithms which displayed a tendency to become stuck in local minima. Ultimately, the results of the paper illustrate the potential of using a PSO algorithm to obtain an accurate dynamic characterization of a TRMS.

A PSO algorithm is also utilized in [2] to identify unknown parameters of a kinematic state-space quadrotor model. The model is separated into a translational subsystem and a rotational subsystem with a PSO algorithm identifying parameters for each subsystem. The identified parameters are then compared to the true values of the parameters, confirming that the PSO algorithm was able to identify parameters within a minimal margin of error.

While few papers have been written about the optimization of aerospace system parameters using the *fmincon* algorithm, [3] utilizes the algorithm for UAV path optimization. The study compares the performance of numerical solvers such as *fmincon* with a mixed-integer nonlinear program, finding *fmincon* to perform about as effectively as the mixed-integer nonlinear program while requiring fewer computational resources.

Throughout these studies, however, the modeling of electrical components and optimization of electrical parameters have been largely absent. As these components and parameters are critical to the behavior of such systems, their inclusion was of particular interest for this work.

## III. Multi-Domain Modeling Approach

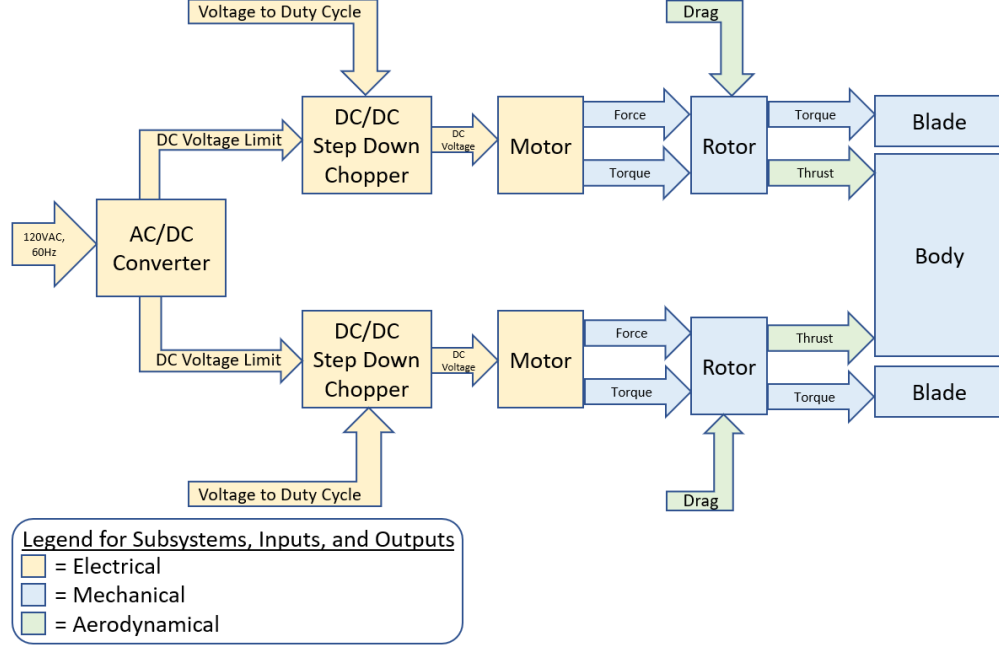
An object-oriented, multi-domain Modelica model of the Quanser AERO was developed in Dymola utilizing components from the Modelica Drone Library [4, 5]. The Modelica modeling language allows for the implementation of models with components from multiple engineering domains by defining equation-based interfaces between components that conform to the principles and laws that govern those domains. The Quanser AERO contains electrical, mechanical, and aerodynamical components that interface with each other as described by the conceptual block diagram shown in Fig. 2.

The electrical components of the AERO include power electronics and the DC motors of the thruster assemblies. The DC motors of the AERO are modeled according to Eqs. 1, 2, and 3. The motor voltage in Eq. 1 is a function of voltage loss due to the armature resistance, armature inductance, and back EMF. The back EMF is a function of a mechanical torque constant and the speed of the motor in Eq. 2. The electrical torque produced by the motor is calculated using Eq. 3.

$$V_m = R_a i_m + L_a \frac{di_m}{dt} + E \quad (1)$$

$$E = c_m \omega \quad (2)$$

$$\tau_e = J_t \frac{d\omega}{dt} \quad (3)$$



**Fig. 2 Block diagram of multi-domain Quanser AERO model.**

The mechanical components of the AERO include the rotor and blades of the thruster assemblies as well as the forces and torques produced by the thruster assemblies. The electrical torque produced by the motor is applied to the rotor. The aerodynamical components of the AERO include the drag and thrust forces it is subject to. An aerodynamic thrust is applied to the rotor block as determined by Eq. 4. A drag force is applied to the blades according to Eq. 5. The thrust and drag coefficients are determined according to the surrounding fluid density, propeller cross sectional area, radius of the propeller, and other constants.

$$F_T = k_t \omega^2 \quad (4)$$

$$F_D = k_d \omega^2 \quad (5)$$

## IV. Optimization Problem Formulation and Tools

### A. RaPIId Overview

Rapid Parameter Identification (RaPIId) is an open-source, modular toolbox developed in MATLAB and Simulink for automated model validation, calibration, and parameter estimation [6]. Its compatibility with Modelica models is made possible by Functional Mock-up Units (FMU) compliant with the Functional Mock-up Interface (FMI) standard\*[7]. When exported as a model exchange FMU, the FMU consists of a model compiled as C code that can be simulated as one cohesive block in Simulink using the Simulink's solver, as shown in Figure 4. While MATLAB and Simulink offer native support for the import of FMUs, RaPIId instead utilizes Modelon's FMI Toolbox [8] to import, simulate, and perform other computations using FMUs. Fig. 3 shows a Simulink model configured for use with the RaPIId toolbox, where the FMU of the Quanser AERO is labeled as *FMUme*.

Using user-defined parameters from a Modelica model imported into Simulink as a FMU, the toolbox simulates and calibrates the model through an iterative process. An optimal set of parameters is determined by minimizing the cost function using the error between the simulation results and reference time-series data. Each point in the parameter identification process in the RaPIId toolbox as depicted in Figure 5 is defined as follows:

\*The FMI is a standard maintained as a Modelica Association Project that defines a container and an interface to exchange dynamic models between tools through a single file.

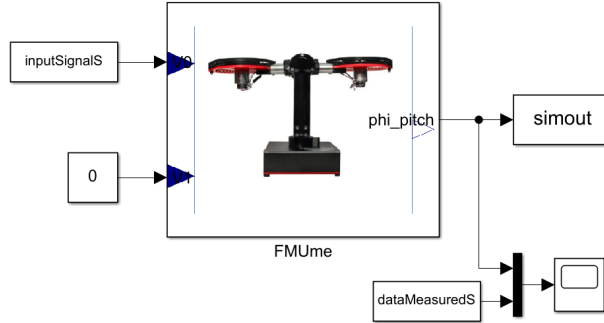


Fig. 3 Simulink model configured to use a Quanser AERO FMU with the RaPIId toolbox.

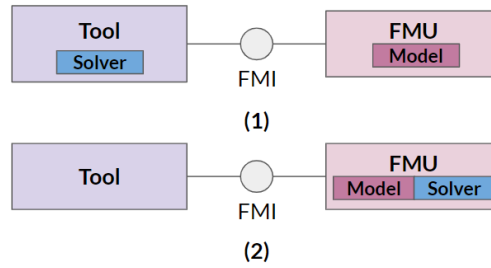


Fig. 4 (1) Link between FMU and other tools for model exchange (2) Link between FMU and other tools for co-simulation

- 1) A reference measurement in the form of time-series data is provided to the toolbox.
- 2) Through a GUI or a MATLAB script, the RaPIId container specifies the methods for parameter identification to be used, the optimization method and its parameters, the numerical solver, and other variables to simulate the FMU in Simulink. Optimization algorithms can be selected from those developed for RaPIId, methods provided by MathWorks, or other third-party tools. It is also possible to combine optimization methods. For example, RaPIId can find a global solution with a PSO before switching to a gradient-based method to refine the solution with minimal computational effort
- 3) An FMU block containing the Modelica model using model exchange in Simulink [9]. When using FMUs exported with the model exchange (ME) feature, any Simulink solver can be used.
- 4) Error is determined by comparing the reference measurement to the simulation results.
- 5) RaPIId uses the error to determine new parameters using the specified optimization method.

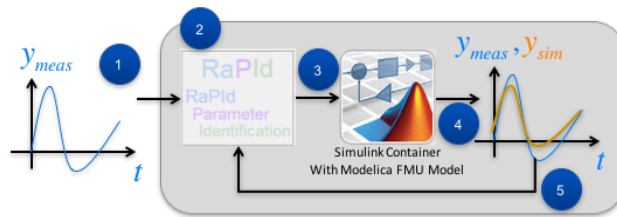


Fig. 5 The RaPIId parameter identification process.

## B. Optimization Problem

An overview of the optimization routine for RaPIId is given in Algorithm 1. Regardless of the selected optimization algorithm, vector  $p_0$  contains the initial values for the parameters of interest as specified by the vector `parameterNames`. The vectors of upper and lower parameter boundaries,  $p_{max}$  and  $p_{min}$  respectively, have been determined based on

the error tolerance of the physical components associated with the parameters of interest. The measurement data in `experimentData` is comprised of time vectors and measurement vectors for the inputs to and outputs from the FMU. While the RaPID toolbox includes numerous optimization algorithms to select from, this work focuses on comparing the efficacy of a PSO algorithm and a constrained nonlinear optimization algorithm in the optimization of several electrical, mechanical, and aerodynamical parameters of interest for a Quanser AERO model.

---

**Algorithm 1** Parameter estimation using RaPID

---

**Define simulation experiment:**

$p_i = i^{th}$  vector of values for all parameters to be calibrated  
 $p_{min}$  = vector of lower limit of all parameters calibrated  
 $p_{max}$  = vector of upper limit of all parameters calibrated  
`experimentSettings` = path to Simulink model, FMU, integration method, and other simulation settings.  
`experimentData` = path to vectors of reference measurement data and input signal data  
`cost_function` = integer to select cost function to calculate error  
 $\epsilon_0$  = error tolerance  
`optimizationAlgorithm` = string to select optimization algorithm

**Define FMU parameters, inputs, and outputs:**

`parameterNames` = vector of names of the parameters of interest to be updated in FMU  
`fmInputNames` = vector of names of inputs to FMU (from measurements)  
`fmOutputNames` = vector of names of outputs from the FMU (to compare with measurements)

**Run parameter estimation with RaPID:**

- 1) Find a solution using the selected optimization algorithm with initial parameters  $p_i$ .
  - 2) Compute the error,  $\epsilon$ , by using the cost function defined during initialization to compare the simulation results to the reference measurement data.
  - 3) If  $\epsilon$  is below  $\epsilon_0$  or the user-defined maximum number of iterations have been achieved, then the parameter set obtained is considered to be the solution. If not, the parameter vector  $p_{i+1}$  is then used as the input for the next iteration of the experiment.
- 

*1. Particle Swarm Optimization*

PSO is a population-based algorithm in which a collection of candidate solutions, or particles, are pseudo-randomly placed in a parameter space according to the bounds specified by  $p_{min}$  and  $p_{max}$ . With every iteration of the optimization routine, the cost function for each particle is evaluated, determining the new velocity of each particle to the next step. The fitness of the model for the specified parameters is then calculated as the sum of the Frobenius norm of the error,  $\epsilon$  at each time step, as described by the cost function in Eq. 6. The goal of the PSO algorithm is to find a solution that is the global minimum, however this is not necessarily a guaranteed outcome. The closer the fitness of a model is to zero, the closer the output of that model fits the measured output it is being compared with. The optimization process terminates if the fitness as determined by this cost function is less than the value defined in `experimentSettings` or if the maximum number of iterations defined in `experimentSettings` is reached.

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n (\epsilon_{ij} * \epsilon_{ij}) \quad (6)$$

*2. Constrained Nonlinear Optimization*

`fmincon` is a constrained nonlinear optimization algorithm included in MathWorks' Optimization Toolbox [10] that finds the minimum of a problem specified by Eq. 7.

$$\min_x f(x) \text{ such that } \left\{ p_{min} \leq x \leq p_{max} \right\}. \quad (7)$$

The error,  $\epsilon$ , is defined as the absolute difference between the measured output and the simulated output calculated at each time step according to Eq. 8. If  $\epsilon$  exceeds the error tolerance as defined in `experimentSettings`, then the resulting parameter vector is used as the initial parameter vector for the next iteration. As before, the fitness of the model for the specified parameters is then calculated as the sum of the Frobenius norm of  $\epsilon$  at each time step, as described by

the cost function in Eq. 6 with the optimization process terminating when the fitness is less than the user-defined value or if the maximum number of iterations is reached.

$$\epsilon = \left[ y_{out}^{simulated} - y_{out}^{reference} \right]^T \quad (8)$$

Approximate values for several parameters of the AERO can be obtained from documentation provided by Quanser [11], therefore it is not necessary to find a global solution. As such, the `fmincon` algorithm is expected to obtain adequate results faster than the PSO algorithm while using fewer computational resources.

## V. Results

The results of three tests optimizing the parameters associated with the armature inductance and resistance, rotor and stator inertia, and drag and thrust coefficients of the Quanser can be seen in Table 1. For all three tests, the input signal provided a constant 24V for 20 seconds before stepping down to 0V for an additional 20 seconds. The electrical parameters were optimized using the current through the motor of the AERO as the measured output data. Similarly, the mechanical parameters were optimized using the mechanical speed of the AERO motors, and the aerodynamical parameters were optimized using the pitch angle of the AERO body. The parameters values for the original multi-domain model were obtained from documentation provided by Quanser [11]. It can be observed from Table 1 that while the original, unoptimized multi-domain model's fitness value of 29.3903 indicates a closer fit to the measured data than the transfer function model's fitness value of 42.7060, the `fmincon` optimized model represents the closest fit to the measured output data of all models with a fitness value of 0.2280. In comparing the fitness values of the two PSO optimized models, it may be inferred that the electrical parameters were poorly optimized by the PSO algorithm, as by replacing the values of the electrical parameters with the unoptimized values, the fitness of the model noticeably improves, slightly outperforming the unoptimized multi-domain model. Note that the transfer function model provided by Quanser does not explicitly consider several parameters, significantly affecting the model's fitness compared to the reference output.

A plot of the measured output of the AERO, the unoptimized multi-domain model, the PSO and `fmincon` optimized models, and the transfer function model provided by Quanser is presented in Figure 6. From this plot, it can qualitatively be observed that the pitch angle output of the `fmincon` optimized model tracks the measured output data more closely than any of the other models. Additionally, in comparing both PSO optimized models, the improvement in fitness after replacing the electrical parameter values with their unoptimized values can qualitatively be observed.

## VI. Conclusion

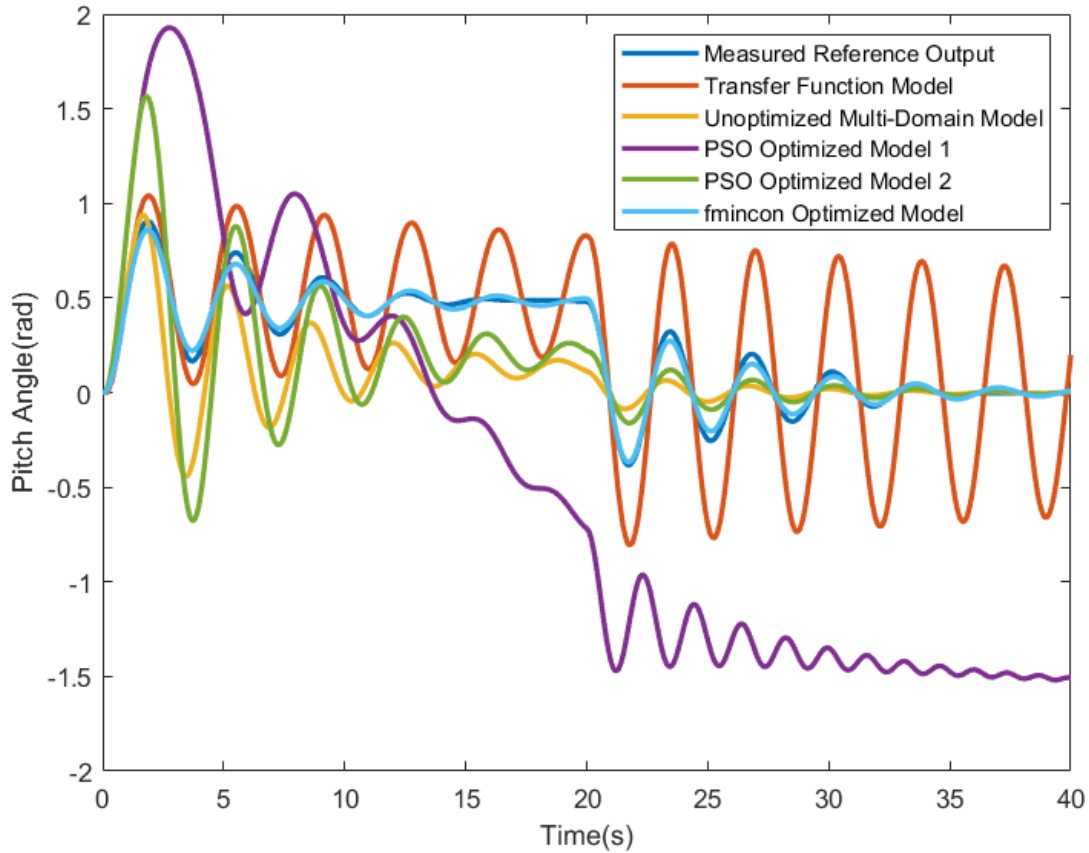
This paper presents an overview of the automated optimization process for specified parameters of a multi-domain model of the Quanser AERO using RaPID. As approximate parameter values for the AERO were available in the documentation provided by Quanser, finding a global solution was unnecessary. As such, the constrained nonlinear optimization algorithm, `fmincon`, was able to find a more adequate local solution in less time than the PSO algorithm.

Future work with the Quanser AERO will expand the modelling of its power source and motors to improve its adequacy and support the further optimization of its electrical and mechanical parameters.

## Appendix

**Table 1 Comparison of model parameters and their fitness.**

| Source  | Parameter | Value    | Units           | Description                          | Fitness  |
|---|-----------|----------|-----------------|--------------------------------------|----------|
| Unoptimized<br>Multi-Domain<br>Model          | $R_a$     | 8.4      | $\Omega$        | Armature resistance of motor         | 29.3903  |
|   | $L_a$     | 0.00116  | H               | Armature inductance of motor         |          |
|   | $J_r$     | 4e-6     | $kg \cdot m^2$  | Moment of inertia of DC motor rotor  |          |
|   | $J_s$     | 4e-6     | $kg \cdot m^2$  | Moment of inertia of DC motor stator |          |
|   | $k_d$     | -1e-5    | $N \cdot s/rad$ | Aerodynamic drag coefficient         |          |
|   | $k_t$     | 0.00045  | $N \cdot s/rad$ | Aerodynamic thrust coefficient       |          |
| PSO<br>Optimized<br>Multi-Domain<br>Model 1   | $R_a$     | 0.187    | $\Omega$        | Armature resistance of motor         | 498.7173 |
|   | $L_a$     | 0.0487   | H               | Armature inductance of motor         |          |
|   | $J_s$     | 0        | $kg \cdot m^2$  | Moment of inertia of DC motor stator |          |
|   | $J_r$     | 84.2     | $kg \cdot m^2$  | Moment of inertia of DC motor rotor  |          |
|   | $k_d$     | -0.587   | $N \cdot s/rad$ | Aerodynamic drag coefficient         |          |
|   | $k_t$     | 0.000689 | $N \cdot s/rad$ | Aerodynamic thrust coefficient       |          |
| PSO<br>Optimized<br>Multi-Domain<br>Model 1   | $R_a$     | 8.4      | $\Omega$        | Armature resistance of motor         | 28.2230  |
|   | $L_a$     | 0.00116  | H               | Armature inductance of motor         |          |
|   | $J_s$     | 0        | $kg \cdot m^2$  | Moment of inertia of DC motor stator |          |
|   | $J_r$     | 84.2     | $kg \cdot m^2$  | Moment of inertia of DC motor rotor  |          |
|   | $k_d$     | -0.587   | $N \cdot s/rad$ | Aerodynamic drag coefficient         |          |
|   | $k_t$     | 0.000689 | $N \cdot s/rad$ | Aerodynamic thrust coefficient       |          |
| fmincon<br>Optimized<br>Multi-Domain<br>Model | $R_a$     | 8.4      | $\Omega$        | Armature resistance of motor         | 0.2280   |
|   | $L_a$     | 0.00116  | H               | Armature inductance of motor         |          |
|   | $J_r$     | 3.45e-11 | $kg \cdot m^2$  | Moment of inertia of DC motor rotor  |          |
|   | $J_s$     | 0.000174 | $kg \cdot m^2$  | Moment of inertia of DC motor stator |          |
|   | $k_d$     | -1e-5    | $N \cdot s/rad$ | Aerodynamic drag coefficient         |          |
|   | $k_t$     | 0.0004   | $N \cdot s/rad$ | Aerodynamic thrust coefficient       |          |
| Transfer<br>Function<br>Model                 | $R_a$     | 8.4      | $\Omega$        | Armature resistance of motor         | 42.7060  |
|   | $k_d$     | -1e-5    | $N \cdot s/rad$ | Aerodynamic drag coefficient         |          |
|   | $k_t$     | 0.00045  | $N \cdot s/rad$ | Aerodynamic thrust coefficient       |          |



**Fig. 6 Pitch angle over time of Quanser AERO and its models.**

### Acknowledgments

This work was supported in whole or in part by the National Aeronautics and Space Administration through the University Leadership Initiative Award Number 80NSSC19M0125 for the Center for High-Efficiency Electrical Technologies for Aircraft (CHEETA), and in part by the Center of Excellence for NEOM Research at the King Abdullah University of Science and Technology under grant OSR-2019-CoE-NEOM-4178.12.

The second author is supported through the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1744655 and the Chateaubriand Fellowship of the Office for Science & Technology of the Embassy of France in the United States.

### References

- [1] Toha, S. F., Latiff, I. A., Mohamad, M., and Tokhi, M. O., "Parametric Modelling of a TRMS Using Dynamic Spread Factor Particle Swarm Optimisation," *2009 11th International Conference on Computer Modelling and Simulation*, 2009, pp. 95–100. <https://doi.org/10.1109/UKSIM.2009.109>.
- [2] Yang, L., and Liu, J., "Parameter identification for a quadrotor helicopter using PSO," *52nd IEEE Conference on Decision and Control*, 2013, pp. 5828–5833. <https://doi.org/10.1109/CDC.2013.6760808>.
- [3] Ragi, S., and Mittlmann, H. D., "Mixed-integer nonlinear programming formulation of a UAV path optimization problem," *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 406–411.
- [4] Podlaski, M., Vanfretti, L., Nademi, H., and Chang, H., "UAV Dynamic System Modeling and Visualization using Modelica and FMI," *76th Annual Forum & Technology Display*, Vertical Flight Society, Virginia Beach, VA, USA, 2020.



- [5] ALSETLab, “Modelica-Drone-3D-FMI,” Website, 1 2020. Url: <https://github.com/ALSETLab/Modelica-Drone-3D-FMI>.
- [6] Vanfretti, L., Baudette, M., Amazouz, A., Bogodorova, T., Rabuzin, T., Lavenius, J., and Gómez-López, F. J., “RaPId: A modular and extensible toolbox for parameter estimation of Modelica and FMI compliant models,” *SoftwareX*, Vol. 5, 2016, pp. 144 – 149. <https://doi.org/https://doi.org/10.1016/j.softx.2016.07.004>, URL <http://www.sciencedirect.com/science/article/pii/S235271101630019X>.
- [7] Modelica Association Project, “FMI Standard,” <https://fmi-standard.org/>, Oct. 2020.
- [8] Modelon, A., “FMI toolbox for Matlab,” URL <http://www.modelon.com/products/fmi-toolbox-for-matlab>, 2014.
- [9] Henningsson, M., Akesson, J., and Tummescheit, H., “An FMI-Based Tool for Robust Design of Dynamical Systems,” *Proceedings of the 10th International Modelica Conference*, Linkoping Electronic Conference Proceedings, Vol. 96, Linkoping University Electronic Press, Lund, Sweden, 2014, pp. 35–42.
- [10] MathWorks, “Optimization Toolbox,” , 2020.
- [11] Quanser, “User Manual: Quanser AERO Experiment,” PDF, 2016.