



**HAL**  
open science

## CW-Quad Toolbox (User's guide)

Gabriele Perozzi

► **To cite this version:**

| Gabriele Perozzi. CW-Quad Toolbox (User's guide). 2018. hal-01696347

**HAL Id: hal-01696347**

**<https://inria.hal.science/hal-01696347>**

Preprint submitted on 15 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# CW-Quad Toolbox (User's guide)

Gabriele Perozzi

February 15, 2018

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Description</b>	<b>2</b>
2.1	Requirements . . . . .	2
2.2	Input . . . . .	3
2.3	Control . . . . .	4
2.4	Quadrotor models . . . . .	5
2.5	Rotors model . . . . .	7
2.6	Sensors . . . . .	8
2.7	Complementary filter . . . . .	9
2.8	Wind estimator . . . . .	9
2.9	Results . . . . .	10
<b>3</b>	<b>Example case</b>	<b>10</b>

# 1 Introduction

CW-Quad Toolbox (Controls: sliding mode controls,  $H_\infty$  control, Pid control; Wind estimator for **Quadrotors**) consists in a group of Simulink libraries to simulate a quadrotor under wind perturbation. This toolbox gathers control theory and aerodynamic science and frees the user to build his/her own model. Fully configurable masks are provided to simulate quadrotors with different shapes and in different environment conditions. This toolbox aims to help the user to study quadrotors in total: trajectories, controls (Sliding mode control, PID,  $H_\infty$ ), state filtering and wind estimator. Only blocks provided with basic Simulink environment are used to ensure a better compatibility with most of Simulink versions, avoiding any other auxiliary toolbox which typically are sold separately.

- **How to use this guide:** This user's guide and the toolbox were built to be scalable: users can easily take and/or modify single simulink subsystem blocks to improve them or to use them in other projects without compromise the whole code, each block has its own detailed description in the following paragraphs, which can be read separately. Each main block has a configurable mask, to look after the mask and to take single subsystems, click on the icon enlightened in Figure 1.

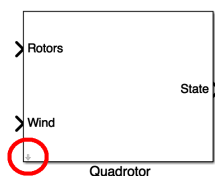


Figure 1: How to look under the masks

- **Note 1:** This guide is an extension of a previous guide: G.Perozzi. A toolbox for quadrotors: from aerodynamic science to control theory. January 2018.

# 2 Description

This toolbox is structured as in the following Figure 2.

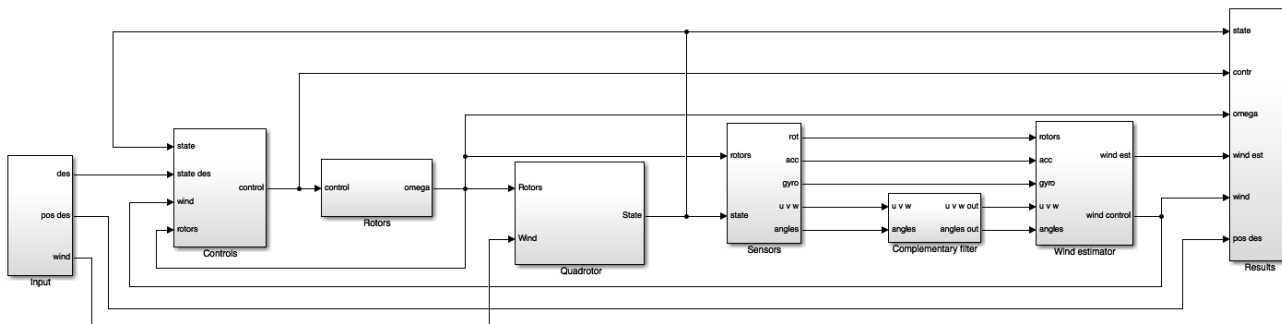


Figure 2: Toolbox scheme

**Remark:** A big attention has been given building the 'Controls', 'Rotors', 'Quadrotor', 'Sensors' blocks, using only Simulink. In this way, it is easier to take and/or modify each single model component without compromise the other code. In opposite to use both Matlab and Simulink which allows to build the model easier and faster, but hard scalable (see Fig. 3).

## 2.1 Requirements

Toolbox created using Matlab/Simulink r2016a. Simulink required blocks are explained in each subsection for the respective toolbox subsystem.

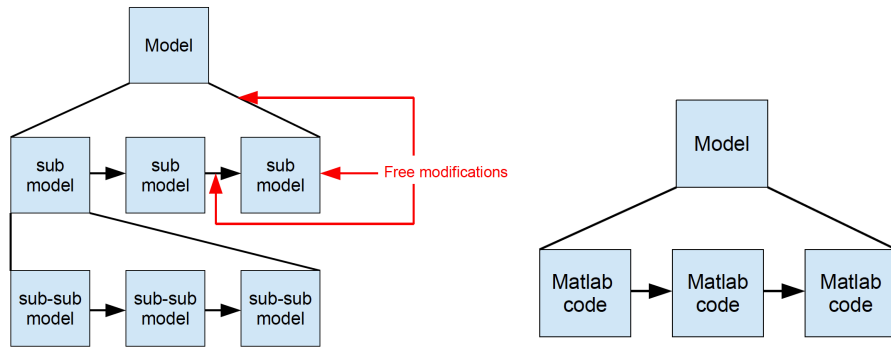


Figure 3: Easy scalable and readable (on the left); hard scalable and readable (on the right) models.

## 2.2 Input

Simulink blocks used are: Constant, Gain, Discrete-Time Integrator, In, Out, Sum, Mux, Demux, Signal builder, Sine wave, Rate transition. The input block is structured as in the following Figure 4.

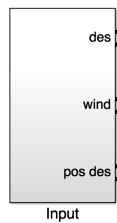


Figure 4: Input block

Output of the model is defined as:

- Output:
  - des: One vector of 12 components of the desired state. In order: 3 position in earth frame expressed in  $m$ , 3 linear velocities in earth frame expressed in  $m/s$ , 3 linear accelerations in earth frame expressed in  $m^2/s$ .
  - pos des: One vector of 3 position in earth frame expressed in  $m$ .
  - wind: One vector of 3 components of the wind velocity in earth frame expressed in  $m/s$ .

The following parameter values can be changed manually:

- $x, y, z$  step desired signals for the quadrotor trajectory in earth frame
- $\psi$  angle in degrees.
- $x, y, z$  wind velocity components in earth frame expressed in  $m/s$ .

The trajectory generation block, which is a subsystem of the input block, is structured as in the following Figure 5.

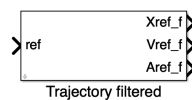


Figure 5: Trajectory filtering block

Input and Output of the model are defined as:

- Input:
  - ref: Reference vector of 3 components  $x, y, z$  in earth frame expressed in  $m$ .
- Output:
  - $Xref_f$ : Vector of 3 components  $x, y, z$  filtered expressed in  $m$ .
  - $Vref_f$ : Vector of 3 components  $\dot{x}, \dot{y}, \dot{z}$  velocities filtered expressed in  $m/s$ .
  - $Aref_f$ : Vector of 3 components  $\ddot{x}, \ddot{y}, \ddot{z}$  accelerations filtered expressed in  $m/s^2$ .

The following parameter values can be changed:

- Quadrotor limitations:
  - Maximal and minimal acceleration;
  - Maximal and minimal velocity;
  - Maximal and minimal position.
- Weighted constants for the chosen filtered trajectory:
  - $\tau_{pos}$ : To influence the trajectory  $x, y, z$ ;
  - $\tau_{spe}$ : To influence the trajectory  $\dot{x}, \dot{y}, \dot{z}$ ;
  - $\tau_{acc}$ : To influence the trajectory  $\ddot{x}, \ddot{y}, \ddot{z}$ .

## 2.3 Control

Simulink blocks used are: Constant, In, Out, Index vector, Mux, Demux, Unary minus, Saturation, Product, Add, Trigonometric function, Subtract, Gain, Fcn, Abs, Initial condition, Math function, MinMax, Reshape, Scope, Sqrt, Divide, Abs. The control block is structured as in the following Figure 6.

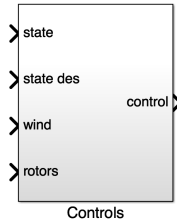


Figure 6: Controls block

Input and Output of the model are defined as:

- Input:
  - state: One vector of 15 components of the state of the system. In order: 3 linear velocities in earth frame expressed in  $m/s$ , 3 angular velocities expressed in  $rad/s$ , 3 angles expressed in  $rad$ , 3 position in earth frame expressed in  $m$ , 3 linear accelerations in earth frame expressed in  $m/s^2$ .
  - rotors: vector of 4 components. Rotors angular velocity in  $rad/s$ .
  - wind: vector of 3 estimated wind velocity components in  $x, y, z$  directions expressed in body frame with respect to the earth and in  $m/s$ .
  - state des: vector of 9 components, in order: 3 desired  $x, y, z$  position in earth frame expressed in  $m$ ; 3 desired velocity in earth frame expressed in  $m/s$ ; 3 desired accelerations in earth frame expressed in  $m/s^2$ .
- Output:
  - control: vector of 4 components,  $U_z, U_\phi, U_\theta, U_\psi$ , expressed respectively in  $N, Nm, Nm, Nm$ .

The following parameter values can be changed:

- Choose the desired control.
- Choose the values for the quasi-continuous SMC (qc-SMC).
- Choose the values for the first order SMC (1-SMC).
- Choose the values for the Proportional Integrative Derivative control (PID).

## 2.4 Quadrotor models

The quadrotor block is structured as in the following Figure 7.

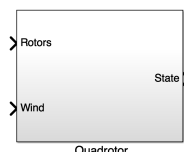


Figure 7: Quadrotor block

Two quadrotor models are implemented:

- A nonlinear model which includes all the nonlinear aerodynamic coefficients, called 'complex'.
- A nonlinear model which includes simplifications based on an identification work, extensively explained in the previous guide (see Note 1) and called 'simple'.

The best strategy to work with these models is to use at first the simple one, which is more suitable to be studied for control and estimation techniques, then validate the algorithms with the complex model, checking the differences due to the neglected nonlinear terms. Their peculiarities are that:

- Only basic simulink blocks are used, avoid the necessity of interpreted or external matlab functions, allowing better compatibility with most of simulink versions. In particular the used blocks are: Constant, Mux, Demux, Gain, Integrator, In, Out, Product, Terminator, Switch, Compare To Constant, Relational Operator, Logical Operator, Abs, Add, Divide, Math Function, Product of Elements, Sqrt, Subtract, Sum of Elements, Unary Minus, Unit Delay (it is only used in the complex model), Trigonometric Function, Subsystem, While Iterator Subsystem (it is only used in the complex model), From, Goto, Stop Simulation, To Workspace, Cross vector.
- Auxiliary matlab/simulink toolbox, such as the purchasable Aerospace toolbox provided by Simulink, is not needed except of the basic simulink package.
- They are based on a detailed model ensuring a good quality simulator close to the reality.
- They are fully configurable changing the parameter values, allowing to study any kind of quadrotor having shape similar to the commercial Parrot AR.Drone 2.0.
- The complex model can recognize the vortex ring state and stop the simulation accordingly. When the variable "test", in matlab workspace, is equal to 1.

Input and Output of the model are defined as:

- Input:
  - Rotors velocity: One vector of angular velocities expressed in  $rad/s$ , where the 4 rotors are indicated with their signs according to the opposite direction, ex: +, -, + -.
  - Wind: One vector of 3 components of the wind velocity in earth frame expressed in  $m/s$ .
- Output:

- State: One vector of 15 components of the state of the system. In order: 3 linear velocities in earth frame expressed in  $m/s$ , 3 angular velocities expressed in  $rad/s$ , 3 angles expressed in  $rad$ , 3 position expressed in  $m$ , 3 linear accelerations in earth frame expressed in  $m/s^2$ .

The following parameter values can be changed to simulate quadrotors with different shapes and in different environment conditions using the highly nonlinear coefficients:

- Environment:
  - Gravity acceleration.
  - Air density.
- Quadrotor body:
  - Mass of the drone.
  - Moment of Inertia on x axes.
  - Moment of Inertia on y axes.
  - Moment of Inertia on x-z axes.
  - Angle  $\epsilon$ .
  - Arm length.
  - Distance between the center of gravity of the drone and the rotors plane.
- Rotors:
  - Angle of attack of the blade at the root profile.
  - Twist of the blade.
  - Rotor radius.
  - Lift curve slope of the blade section.
  - Blades number.
  - Main chord length of the blades.
  - Drag coefficient of the blade section.
  - Induced drag coefficient of the blade section.
- Blade flapping equation:
  - Position of the fictitious spring.
  - Center of gravity of the blade.
  - Mass of the blade.
  - Inertia of the blade.
- Auxiliary parameter to solve the  $\lambda$  equation:
  - Maximum number of allowed iterations to solve the nonlinear  $\lambda$  equation, after that the system is in unstable position and automatically blocks the simulation: It means we are in vortex ring state.
- UAV initial conditions:
  - Initial linear velocity in earth frame.
  - Initial position in earth frame.
  - Initial angular velocity.
  - Initial angles.

The following parameter values can be changed to simulate quadrotors with different shapes and in different environment conditions using the simplifying hypothesis based on identification:

- Environment:

- Gravity acceleration.
- Air density.
- Quadrotor body:
  - Mass of the drone.
  - Moment of Inertia on x axes.
  - Moment of Inertia on y axes.
  - Angle  $\epsilon$ .
  - Arm length.
  - Distance between the center of gravity of the drone and the rotors plane.
- Rotors:
  - Angle of attack of the blade at the root profile.
  - Rotor radius.
  - Lift curve slope of the blade section.
  - Blades number.
  - Main chord length of the blades.
  - Rotors drag coefficient ( $K_D$ ).
  - Rotors thrust coefficient ( $K_z$ ).
  - Rotors thrust coefficient in stationary phase ( $C_{Tstat}$ ).
- UAV initial conditions:
  - Initial linear velocity in earth frame.
  - Initial position in earth frame.
  - Initial angular velocity.
  - Initial angles.

## 2.5 Rotors model

An auxiliary rotor model is also provided with a configurable mask, which allows to transform the control input to the rotors velocity taking into account the rotor transfer function. Simulink blocks used are: Constant, Sqrt, Product, Divide, Abs, Gain, Discrete-Time Integrator, In, Out, Sum. The rotors model block is structured as in the following Figure 8.

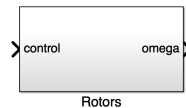


Figure 8: Rotors block

Input and Output of the model are defined as:

- Input:
  - control: vector of 4 components,  $U_z, U_\phi, U_\theta, U_\psi$ , expressed respectively in  $N, Nm, Nm, Nm$ .
- Output:
  - omega: vector of 4 components. Rotors angular velocity in  $rad/s$ .

The following parameter values can be changed to simulate rotors with different shapes and in different environment conditions using the mixer matrix:



- Choose if you want to use the rotors dynamics.
- Rotor transfer function:
  - Denominator.
  - Numerator.
  - Rotor angular velocity in hover phase.
- Mixer Matrix:
  - Air density.
  - Rotor radius.
  - Rotor thrust coefficient in hover flight ( $C_{Tstat}$ ).
  - Drag coefficient of the blade section.
  - Angle of attack at the root profile of the blade section.
  - Rotors position ( $\epsilon$ ).
  - Blades number.
  - Main chord length of the blade section.
  - Drone arm length.

## 2.6 Sensors

Simulink blocks used are: In, Out, Mux, Demux, Add, Cross vector, Math function, Product, Random number, Constant, Reshape, Fcn. The sensors block is structured as in the following Figure 9.

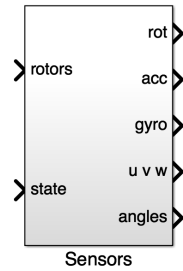


Figure 9: Sensors block

Input and Output of the model are defined as:

- Input:
  - state: One vector of 15 components of the state of the system. In order: 3 linear velocities in earth frame expressed in  $m/s$ , 3 angular velocities expressed in  $rad/s$ , 3 angles expressed in  $rad$ , 3 position in earth frame expressed in  $m$ , 3 linear accelerations in earth frame expressed in  $m/s^2$ .
  - rotors: vector of 4 components. Rotors angular velocity in  $rad/s$ .
- Output:
  - rot: vector of 4 components. Rotors angular velocity in  $rad/s$ .
  - acc: vector of 3 accelerometer components in  $x, y, z$  directions expressed in  $m/s^2$
  - gyro: vector of 3 gyroscope components around  $\phi, \theta, \psi$  directions expressed in  $rad/s$
  - u v w: vector of 3 components of the quadrotor linear velocity expressed in body frame in  $m/s$ .
  - angles: vector of 3 components of the quadrotor angles  $\phi, \theta, \psi$  expressed in  $rad/s$ .

The following parameter values can be changed to simulate sensors with different noises:

- Variance on the gyroscope expressed in  $rad/s$ .
- Variance on the accelerometer expressed in  $m/s^2$ .

## 2.7 Complementary filter

Simulink block used is: Matlab function. The complementary filter block is structured as in the following Figure 10.

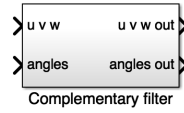


Figure 10: Complementary filter block

Input and Output of the model are defined as:

- Input:
  - u v w: vector of 3 components of the quadrotor linear velocity expressed in body frame and in  $m/s$ .
  - angles: vector of 3 components of the quadrotor angles  $\phi$ ,  $\theta$ ,  $\psi$  expressed in  $rad/s$ .
- Output:
  - u v w out: filtered vector of 3 components of the quadrotor linear velocity expressed in body frame and in  $m/s$ .
  - angles out: filtered vector of 3 components of the quadrotor angles  $\phi$ ,  $\theta$ ,  $\psi$  expressed in  $rad/s$ .

## 2.8 Wind estimator

Simulink blocks used are: In, Out, Math function, Constant, Product, Gain, Add, Subtract, Abs, Sign, Integrator, Matlab function, Divide, From, Reshape, Fcn, Mux, Demux. The wind estimator block is structured as in the following Figure 11.

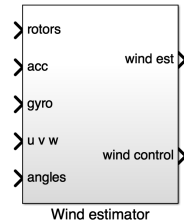


Figure 11: Wind estimator block

Input and Output of the model are defined as:

- Input:
  - rotors: vector of 4 components. Rotors angular velocity in  $rad/s$ .
  - acc: vector of 3 accelerometer components in  $x$ ,  $y$ ,  $z$  directions expressed in  $m/s^2$ .
  - gyro: vector of 3 gyroscope components around  $\phi$ ,  $\theta$ ,  $\psi$  directions expressed in  $rad/s$ .
  - u v w: vector of 3 components of the quadrotor linear velocity expressed in body frame and in  $m/s$ .
  - angles: vector of 3 components of the quadrotor angles  $\phi$ ,  $\theta$ ,  $\psi$  expressed in  $rad/s$ .
- Output:
  - wind est: vector of 3 components of the wind in  $x$ ,  $y$ ,  $z$  directions expressed in earth frame and in  $m/s$ .
  - wind control: vector of 3 components of the wind in  $x$ ,  $y$ ,  $z$  directions expressed in body frame with respect to the earth and in  $m/s$ .

The following parameter values can be changed to use different wind estimator algorithms and to 'adjust' them to different quadrotors:

- Choose the desired algorithm: based on the linear quadrotor dynamics, based on the angular quadrotor dynamics, based on the fusion between linear and angular quadrotor dynamics.
- Parameters for the algorithm using linear quadrotor dynamics.
- Parameters for the algorithm using angular quadrotor dynamics.
- Fusion algorithm:
  - $K_a$ : parameter for the weighted fusion, influencing the the linear quadrotor dynamics algorithm;
  - $K_{g2}$ : parameter for the weighted fusion, influencing the angular quadrotor dynamics algorithm.

## 2.9 Results

Simulink blocks used are: Scope, Gain, In, Mux, Demux. The results block is structured as in the following Figure 12.

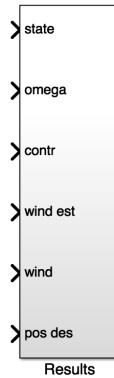


Figure 12: Results block

Input of the model is defined as:

- Input:
  - state: One vector of 15 components of the state of the system. In order: 3 linear velocities in earth frame expressed in  $m/s$ , 3 angular velocities expressed in  $rad/s$ , 3 angles expressed in  $rad$ , 3 position in earth frame expressed in  $m$ , 3 linear accelerations in earth frame expressed in  $m/s^2$ .
  - omega: vector of 4 components. Rotors angular velocity in  $rad/s$ .
  - contr: vector of 4 components,  $U_z, U_\phi, U_\theta, U_\psi$ , expressed respectively in  $N, Nm, Nm, Nm$ .
  - wind est: vector of 3 components of the estimated wind in  $x, y, z$  expressed in earth frame and in  $m/s$ .
  - wind: vector of 3 components of the wind in  $x, y, z$  directions expressed in earth frame and in  $m/s$ .
  - pos des: vector of 3 position components of the filtered trajectory in  $x, y, z$  expressed in earth frame and in  $m$ .

## 3 Example case

One example case is illustrated below. For a better comprehension the user should follow the steps in order.

- **Note 2:** The simulation runs faster if the *scope* graphics, to check the results, are open only at the end.
1. **Set the input**  
Open the input block and choose the desired wind and position way-points in earth frame.
  2. **Choose the trajectory constraints** Open the trajectory filtered mask and choose the desired trajectory constraints.

3. **Choose the control** Open the controls mask and choose the desired control with its parameters.
4. **Choose the rotors dynamics** Open rotors mask and choose the rotors dynamics.
5. **Choose the quadrotor model** Open the Quadrotor mask and choose which model you prefer and the configuration.
6. **Choose the sensors noise** Open the sensors mask and choose the sensors noise.
7. **Choose the wind estimator** Open the wind estimator mask and choose which algorithm to use and its parameters.
8. **Check the results** Open the results block and open the *scopes* to check your final simulation results:
  - **Note:** If the simulation stops before the end, and the scopes don't show any issue. Check the variable *test* in matlab workspace. If its value is 1, it means the  $\lambda$  equation of the quadrotor model cannot be solved.