

## Article

# Quaternion Attitude Control System of Highly Maneuverable Aircraft

Michał Gołabek <sup>\*</sup>, Michał Welcer, Cezary Szczepański , Mariusz Krawczyk, Albert Zajdel  and Krystian Borodacz

Łukasiewicz Research Network—Institute of Aviation, Al. Krakowska 110/114, 02-256 Warszawa, Poland

\* Correspondence: michał.golabek@ilot.lukasiewicz.gov.pl

**Abstract:** In the era of rapid advancements in manned and unmanned aviation and robotics, there is a need for high-performance, robust attitude control of highly maneuverable fixed-wing aircraft, both manned and unmanned (UAVs). This paper presents an extension to research on spacecraft attitude control. The article extends existing concepts and applies them to the control problem of aircraft operating in Earth's atmosphere. First, a general concept of quaternions is presented. Next, the attitude controller's architecture is discussed. The controller synthesis is described using quaternion algebra. The quaternion-based attitude controller is then compared with a classical Euler-based attitude controller. The methodology for comparison and performance evaluation of both controllers is described. Lastly, the results of the simulations and a comparison of the two controllers are presented and discussed. The presented control scheme outperforms classical methods based on Euler angles, particularly at the aircraft's high pitch and roll angles.



**Citation:** Gołabek, M.; Welcer, M.; Szczepański, C.; Krawczyk, M.; Zajdel, A.; Borodacz, K. Quaternion Attitude Control System of Highly Maneuverable Aircraft. *Electronics* **2022**, *11*, 3775. <https://doi.org/10.3390/electronics11223775>

Academic Editors: Imre J. Rudas, Piotr Szymak, Stanisław Hozyń and Paweł Piskur

Received: 7 October 2022

Accepted: 16 November 2022

Published: 17 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The objective of this paper is to present an attitude controller for fixed-wing aircraft operating over large ranges of pitch and roll angles. The presented controller provides desired control qualities over a wide range of flight conditions in terms of speed of response and performance robustness. The aircraft considered in this paper is the Extra 330SC, which is a single-seat aerobatic airplane, manufactured by Extra Flugzeugbau. The long-term goal justifying the need for such a controller is to design a complete autopilot system for unmanned, highly maneuverable fixed-wing strike and fighter aircraft. The attitude controller in this context can be thought of as one of the internal control loops of the autopilot. It means that input to the attitude controller comes from an external controller, such as a guidance system.

Often when speaking about an aircraft's attitude, Euler angles appear in mathematical description. The Euler angles approach is frequently chosen because of its intuitiveness. Although intuitive, Euler angles suffer from non-linearities and singularities, resulting in undesired effects such as gimbal lock. Moreover, an aircraft's control based on Euler angles results in unwanted effects of control coupling between a rudder and an elevator for high roll and pitch angles. This paper presents an effective way of dealing with shortcomings associated with Euler angles by replacing them with quaternions in attitude description and control.

A quaternion number system extends the complex number system and is applied to describe rotations in three-dimensional space. A more stable and often simpler attitude representation can be achieved with quaternions than with Euler angles. Quaternions' computational stability comes from the fact that only unit quaternions are required to describe rotations, reducing numeric errors when they are normalized. They lack the gimbal-lock effect inherent to Euler angles because of their relationship to the axis-angle

representation. Their mathematical simpleness comes from the fact that no trigonometric functions are required, only arithmetic functions such as quaternion products (which are computationally efficient). Mathematical tools associated with quaternions also prove the advantage of quaternions over Euler angles. Linear interpolation between two rotations is straightforward in quaternion notation, while impossible in Euler angles notation. Moreover, integrating angular rates to quaternions is much easier than integrating angular rates to Euler angles. This article will present an efficient use of quaternions in an airplane attitude control problem.

Research in quaternion attitude control has been conducted in the area of spacecraft control, and its results can be found in [1–3]. This article aims to extend this control concept and apply it to the aircraft operating in Earth's atmosphere.

The quaternion number system extends the complex number system. A quaternion is a four-dimensional hyper-complex number that can describe transformations in three-dimensional space. Unit quaternions, being quaternions with a norm equal to one, are often used to describe rotations in three-dimensional space. Publications [4–6] explain the quaternion construct and associated quaternion algebra in more detail.

A quaternion consists of one real part and three imaginary parts.  $q_w$  is called the real part and  $q_x, q_y, q_z$  are called imaginary parts. Equations (1) and (2) present the two most popular quaternion notations.

$$q = q_w + q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k} \quad (1)$$

$$q = [q_w \ q_x \ q_y \ q_z]^T \quad (2)$$

The complex conjugate of a quaternion is defined as:

$$\bar{q} = [q_w \ -q_x \ -q_y \ -q_z]^T \quad (3)$$

The norm of a quaternion  $q$  is defined as:

$$|q| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \quad (4)$$

The unit quaternion is a quaternion of the norm equal to one.

$$|q_{unit}| = 1 \quad (5)$$

The quaternion inverse is defined as:

$$q^{-1} = \frac{\bar{q}}{\|q\|^2} \quad (6)$$

For unit quaternion, its inverse is equal to its conjugate:

$$q_{unit}^{-1} = \bar{q}_{unit} \quad (7)$$

The quaternion product of quaternions  $p$  and  $q$ , also called the Hamilton product and denoted by  $\otimes$ , is defined as:

$$p \otimes q = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (8)$$

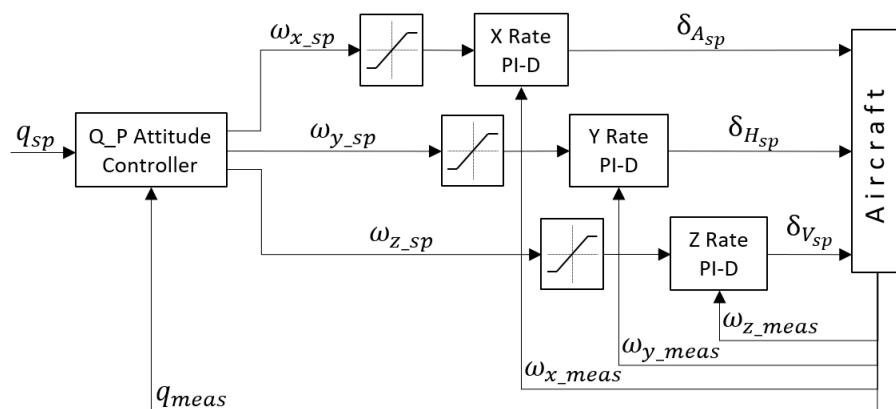
Quaternion multiplication is non-commutative in the same way as real-life rotations are non-commutative:

$$p \otimes q \neq q \otimes p \quad (9)$$

All quaternions used in this paper to represent rotation or attitude are unit quaternions. Please note that this does not apply to quaternion derivatives. Care must be taken when calculating a product of a quaternion and a quaternion derivative since this product will also not be unitary.

## 2. Materials and Methods

The entire attitude control process is performed in the quaternion form. Yaw, pitch and roll angles are not used to control the aircraft's attitude. The classical three controllers (each for a separate: yaw, pitch and roll angle) have been replaced with one controller operating on a single quaternion representing the aircraft's attitude. The proposed controller is comprised of two major parts working in a cascade. The first (external) part consists of a single quaternion "Q\_P Attitude Controller" working similarly to a classical "P" (proportional) controller but being evaluated in quaternion space (Q\_P stands for quaternion proportional). The second (internal) part consists of three PID controllers [7] responsible for angular rate control of the aircraft in its three axes. The inputs to the Q\_P Attitude Controller are: a desired (setpoint) attitude in the form of a quaternion, denoted as  $q_{sp}$ , and a measured attitude in the form of a quaternion, denoted as  $q_{meas}$ . Internally, the controller calculates the attitude error, denoted as  $q_{err}$ , according to Equation (11). The output of the Q\_P Attitude Controller are three angular rate setpoints of the aircraft around its three axes:  $x$ ,  $y$ ,  $z$ , denoted as:  $\omega_{x_{sp}}$ ,  $\omega_{y_{sp}}$ ,  $\omega_{z_{sp}}$ . The output of the Q\_P Attitude Controller (after saturation) constitutes an input to the three angular rate controllers, which are PID controllers [7]. The PID controllers take two inputs. The first input is the setpoint angular rate, and the second is the measured angular rate from the feedback path. The PI part of the controller works on the angular rate error (setpoint minus measured value), and the D part works on the measured value from the feedback path. The outputs from the angular rate controllers are the desired angles of deflection of the aircraft's control surfaces denoted as:  $\delta_{A_{sp}}$ ,  $\delta_{H_{sp}}$ ,  $\delta_{V_{sp}}$ , which correspond to the deflection angles of the ailerons, elevator and rudder, respectively. Feedback signals in the system consist of three measured angular rates around the  $x$ ,  $y$  and  $z$  axes denoted as:  $\omega_{x_{meas}}$ ,  $\omega_{y_{meas}}$ ,  $\omega_{z_{meas}}$ , and a quaternion representing an aircraft's measured attitude  $q_{meas}$ . All signals in the feedback path come from the aircraft's Attitude and Heading Reference System (AHRS) (Figure 1).

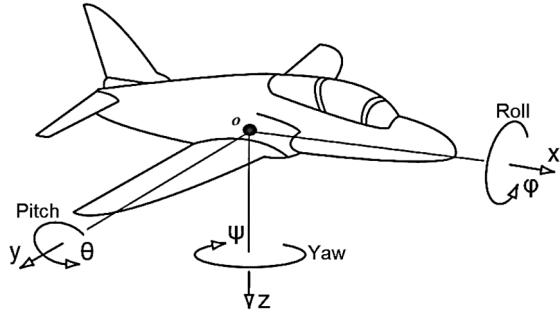


**Figure 1.** The schematic of the quaternion-based controller architecture.

It has to be noted that the presented quaternion attitude control system is not a complete autopilot system. The presented attitude controller constitutes an inner part of the autopilot. An outer part in a form of a guidance system (GS) is required for a complete autopilot. The guidance system is responsible for calculating the setpoint attitude of the aircraft based on assumed flight plan, in particular the assumed flight trajectory and flight parameters. The attitude control system then executes the setpoint attitude calculated by the GS. In this paper, the GS was replaced by man-generated attitude setpoints contained

in test scenarios. In the future, we plan to make GS work fully automatically, which can be achieved, for example, by implementing the inverse dynamics methods described in [8,9].

The assumed coordinate system of the airplane is the standard aeronautical reference system [10], where the  $x$ -axis points to the nose of the aircraft, the  $y$ -axis points to the right wingtip of the aircraft and the  $z$ -axis points to the bottom of the airplane's fuselage (Figure 2).



**Figure 2.** The assumed aircraft coordinate system [10].

This section describes the process of the Q\_P Attitude Controller synthesis. We begin with a definition of the aircraft's attitude error. The attitude error is a difference between a setpoint attitude and a measured attitude. Typically, in a regular PID controller operating on real numbers, an error value is defined as a difference between the desired setpoint (SP) and the measured process variable (PV), i.e.,  $e(t) = SP(t) - PV(t)$  [7]. In the proposed quaternion attitude controller, the attitude error quaternion  $q_{err}$  is calculated based on the following relationship (10) between the setpoint attitude quaternion  $q_{sp}$  and the measured attitude quaternion  $q_{meas}$ , assuming intrinsic rotations:

$$q_{sp} = q_{meas} \otimes q_{err} \quad (10)$$

Thus, after left-multiplication by  $\bar{q}_{meas}$  we obtain:

$$q_{err} = \bar{q}_{meas} \otimes q_{sp} \quad (11)$$

Since two rotations can describe every attitude, we have to ensure that the attitude error  $q_{err}$  is represented by the shorter of the two possible rotations. To do this, we convert a quaternion  $q_{err}$  to axis-angle representation (12). In this representation, we check the angle of rotation  $\theta_{err}$  around the axis  $u$ . If the angle is greater than 180 degrees (13), we invert the axis of rotation and calculate a new angle of the shorter rotation  $\theta_{err,short}$  (14).

$$\begin{aligned} \theta_{err} &= 2 \cdot \text{acos}(q_{werr}) \\ u_x &= \frac{q_x}{\sqrt{1-q_{werr}^2}} \\ u_y &= \frac{q_y}{\sqrt{1-q_{werr}^2}} \\ u_z &= \frac{q_z}{\sqrt{1-q_{werr}^2}} \end{aligned} \quad (12)$$

If:

$$\theta_{err} > \pi \quad (13)$$

then:

$$\begin{aligned} \theta_{err,short} &= |\theta_{err} - 2\pi| \\ u_{x,short} &= -u_x \\ u_{y,short} &= -u_y \\ u_{z,short} &= -u_z \end{aligned} \quad (14)$$

else:

$$\begin{aligned}\theta_{err_{short}} &= \theta_{err} \\ u_{x_{short}} &= u_x \\ u_{y_{short}} &= u_y \\ u_{z_{short}} &= u_z\end{aligned}\quad (15)$$

Next, we need to convert the new attitude error from the axis–angle representation back to the quaternion form (16).

$$\begin{aligned}q_{w_{err_{short}}} &= \cos\left(\frac{\theta_{err_{short}}}{2}\right) \\ q_{x_{err_{short}}} &= u_{x_{short}} \cdot \sin\left(\frac{\theta_{err_{short}}}{2}\right) \\ q_{y_{err_{short}}} &= u_{y_{short}} \cdot \sin\left(\frac{\theta_{err_{short}}}{2}\right) \\ q_{z_{err_{short}}} &= u_{z_{short}} \cdot \sin\left(\frac{\theta_{err_{short}}}{2}\right)\end{aligned}\quad (16)$$

A faster check for the shorter rotation can be performed according to [2].

If:

$$q_{w_{err}} < 0 \quad (17)$$

then:

$$\begin{aligned}q_{w_{err_{short}}} &= -q_{w_{err}} \\ q_{x_{err_{short}}} &= -q_{x_{err}} \\ q_{y_{err_{short}}} &= -q_{y_{err}} \\ q_{z_{err_{short}}} &= -q_{z_{err}}\end{aligned}\quad (18)$$

else:

$$\begin{aligned}q_{w_{err_{short}}} &= q_{w_{err}} \\ q_{x_{err_{short}}} &= q_{x_{err}} \\ q_{y_{err_{short}}} &= q_{y_{err}} \\ q_{z_{err_{short}}} &= q_{z_{err}}\end{aligned}\quad (19)$$

Next, we assume that a quaternion derivative  $\dot{q}_{sp}$  representing a desired (setpoint) rate of rotation required to achieve a desired (setpoint) attitude is proportional to the attitude error, i.e.,  $\dot{q}_{sp} \propto q_{err}$ . After the introduction of proportional gain  $K_p$  the setpoint rate of rotation becomes (20):

$$\dot{q}_{sp} = K_p \cdot q_{err_{short}} \quad (20)$$

Next, we need to express the obtained setpoint rate of rotation as an angular rate vector  $\omega_{sp}$ . The relationship between the quaternion derivative and angular rate vector in the body reference frame is given by [6] (21):

$$\omega = 2\bar{q} \otimes \dot{q} \quad (21)$$

Thus:

$$\omega_{sp} = 2\bar{q}_u \otimes \dot{q}_{sp} \quad (22)$$

where  $q_u$  is an unrotated unit quaternion (23):

$$q_u = [1 \ 0 \ 0 \ 0]^T \quad (23)$$

and its conjugate  $\bar{q}_u$  (24):

$$\bar{q}_u = q_u \quad (24)$$

In extended form, the setpoint angular rate vector becomes (25):

$$\begin{bmatrix} \omega_{x_{sp}} \\ \omega_{y_{sp}} \\ \omega_{z_{sp}} \end{bmatrix} = 2 \cdot \begin{bmatrix} \bar{q}_{w_u} \cdot \dot{q}_{x_u} + \bar{q}_{x_u} \cdot \dot{q}_{w_{sp}} + \bar{q}_{y_u} \cdot \dot{q}_{z_{sp}} - \bar{q}_{z_u} \cdot \dot{q}_{y_{sp}} \\ \bar{q}_{w_u} \cdot \dot{q}_{y_u} - \bar{q}_{x_u} \cdot \dot{q}_{z_{sp}} + \bar{q}_{y_u} \cdot \dot{q}_{w_{sp}} + \bar{q}_{z_u} \cdot \dot{q}_{x_{sp}} \\ \bar{q}_{w_u} \cdot \dot{q}_{z_{sp}} + \bar{q}_{x_u} \cdot \dot{q}_{y_{sp}} - \bar{q}_{y_u} \cdot \dot{q}_{x_{sp}} + \bar{q}_{z_u} \cdot \dot{q}_{w_{sp}} \end{bmatrix} \quad (25)$$

Thus, after simplification (26):

$$\begin{bmatrix} \omega_{x_{sp}} \\ \omega_{y_{sp}} \\ \omega_{z_{sp}} \end{bmatrix} = 2 \cdot \begin{bmatrix} \dot{q}_{x_{sp}} \\ \dot{q}_{y_{sp}} \\ \dot{q}_{z_{sp}} \end{bmatrix} \quad (26)$$

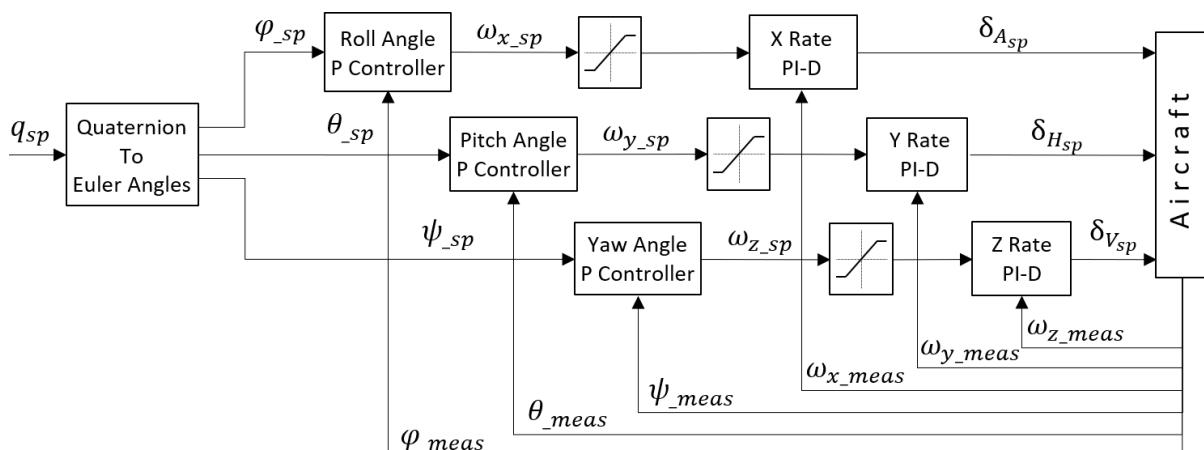
and finally (27):

$$\begin{bmatrix} \omega_{x_{sp}} \\ \omega_{y_{sp}} \\ \omega_{z_{sp}} \end{bmatrix} = 2 \cdot K_p \cdot \begin{bmatrix} q_{x_{err\_short}} \\ q_{y_{err\_short}} \\ q_{z_{err\_short}} \end{bmatrix} \quad (27)$$

The  $\omega_{sp}$  vector is an output from the Q\_P Attitude Controller and input to the three PID controllers responsible for the aircraft's angular rate control, as shown in Figure 1. The simplicity of the result makes the control computationally efficient and makes a direct physical interpretation of the result possible. A fact worth noting is that, unlike Euler-based attitude controllers, the quaternion-based controller will tend to generate a rotation resulting in the shortest path, provided that the aircraft under consideration can achieve the same angular rates in all three (body) axes.

The controller was fully implemented in the "C" language. An extensive quaternion math library was created to support the required quaternion algebra. Preliminary tests were conducted on a simple dynamic model and ran on a simulation in the "C" language. Further tests were conducted using Simulink R2020b combined with Laminar Research X-Plane 11 simulation environment, using a much more complex dynamic model. The controller's compiled "C" files were used as Simulink blocks during simulations. X-Plane flight dynamics and dynamic airplane models were used during the simulations.

To evaluate the performance of the attitude control, the quaternion-based controller (Figure 1) was compared to the conventional Euler-based controller (Figure 3). The Euler-based controller operates on three Euler angles to represent the aircraft's attitude. Therefore, the attitude control is performed on these three angles instead of the one quaternion as it is accomplished in the quaternion-based controller.

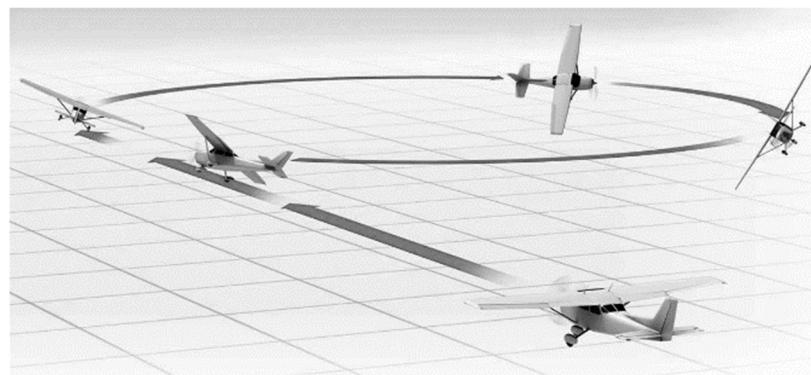


**Figure 3.** The schematic of the Euler-based controller architecture.

The described Euler-based controller is designed in the same cascaded architecture as the proposed quaternion-based controller. The outer control loop is the airplane's attitude

control loop, and the inner control loop is the airplane's angular rate control loop. The outer loop consists of three proportional ("P") controllers. The inner loop consists of three PID controllers, a variation of a classical PID controller, as described in [7]. The outer control loop output (after saturation) constitutes an input to the inner control loop.  $\varphi_{sp}$ ,  $\theta_{sp}$ ,  $\psi_{sp}$  denote setpoint roll, pitch and yaw angles, respectively.  $\omega_{x_{sp}}$ ,  $\omega_{y_{sp}}$ ,  $\omega_{z_{sp}}$  denote setpoint angular rates in the airplane's  $x$ ,  $y$  and  $z$  axes, respectively.  $\delta_{A_{sp}}$ ,  $\delta_{H_{sp}}$ ,  $\delta_{V_{sp}}$  denote commanded deflections of ailerons, elevator and rudder, respectively.  $\omega_{x_{meas}}$ ,  $\omega_{y_{meas}}$ ,  $\omega_{z_{meas}}$  on the feedback path denote measured angular rates in the airplane's  $x$ ,  $y$  and  $z$  axes, respectively.  $\varphi_{meas}$ ,  $\theta_{meas}$ ,  $\psi_{meas}$  on the feedback path denote the airplane's measured roll, pitch and yaw angles, respectively. Details of the classical, Euler-based, cascaded controller can be found in [11]. Computations required to convert from quaternion to Euler angles are described in [5].

Both controllers were implemented in Simulink and coupled with X-Plane 11 simulation environment and then tested against each other during several test scenarios. The test scenarios were designed in such a way as to reflect real-life turn maneuvers of an airplane performed at various roll angles. For example, the turns were simulated at the following roll angles: 30, 60 (Figure 4), 80 and 90 degrees (knife-edge turn).



**Figure 4.** Example of a simulated maneuver—a turn at a 60-degree roll angle.

The test scenarios were designed in the following way: each aircraft's distinct (key) setpoint attitude during the maneuver was given a setpoint "pose" number. The Euler angles ( $\psi$ ,  $\theta$ ,  $\phi$ ) were determined for each distinct pose. Between the following poses, time differences ( $\Delta T$ ) were determined. Test scenarios created this way (Figure 5) were inputs to the Simulink/X-Plane simulation.

	A	B	C	D	E	F
1	Pose #	Time [sec]	Delta T [sec]	Yaw [deg]	Pitch [deg]	Roll [deg]
2	0	0,0		0,0	2,0	0,0
3	1	2,0	2,0	0,0	2,0	0,0
4	2	12,0	10,0	90,0	2,0	60,0
5	3	22,0	10,0	179,0	2,0	60,0
6	4	32,0	10,0	-90,0	2,0	60,0
7	5	42,0	10,0	0,0	2,0	0,0
8	6	102,0	60,0	0,0	2,0	0,0

**Figure 5.** Example of the test scenario—a turn at a 60-degree roll angle.

Each test scenario was processed in Simulink in the following fashion. First, each setpoint pose attitude was converted from Euler angles representation to quaternion representation. Then, intermediate poses were calculated using Spherical Linear Interpolation (SLERP) between all distinct key poses from the scenarios. In this way, a smooth transition between key setpoint poses was obtained. A smooth attitude setpoint was therefore

achieved. This smooth, interpolated attitude setpoint was finally fed to the inputs of the attitude controllers (quaternion-based and Euler-based).

An Extra 330 SC airplane (Figure 6) was chosen to be simulated. A dynamic model of the airplane for the X-Plane 11 was purchased from the official X-Plane Store. Part number of the product: Vertigo-330SC. The fact of its use by one of the Red Bull Air Race professional pilots, during his virtual trainings, validates the accuracy of the purchased dynamic model.



**Figure 6.** Extra 330 SC during the simulated turn.

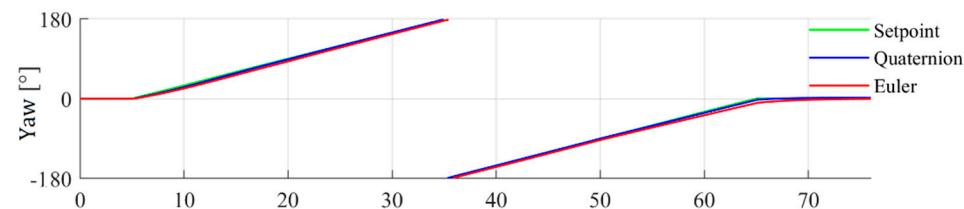
Both attitude controllers utilized equal proportional gains. In addition, both controllers used the same internal loop responsible for angular rate control, so the same PID gains were automatically assured between the two controllers. Therefore, it is important when comparing the performance of the two controllers.

To improve the robustness of the control, the outputs of the angular rate controllers were correlated with the square of the aircraft airspeed, which comes directly from the lift force equation [10].

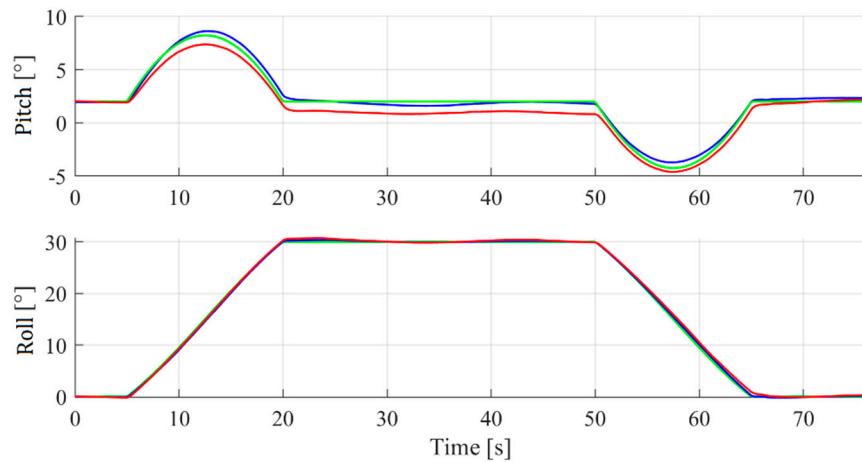
### 3. Results

This section presents the results of the performed simulations. The simulations consisted of turns performed at 30-, 60-, 80- and 90-degree roll angles by the quaternion-based and the Euler-based controllers. For each simulated turn, two types of plots are presented—the setpoint tracking performance plot and the tracking error plot. The setpoint tracking performance plots illustrate the setpoint attitude and the measured attitude achieved by the two controllers, presented in the form of Euler angles ( $\psi, \theta, \varphi$ ). The tracking error plots illustrate tracking errors in yaw, pitch and roll for both controllers.

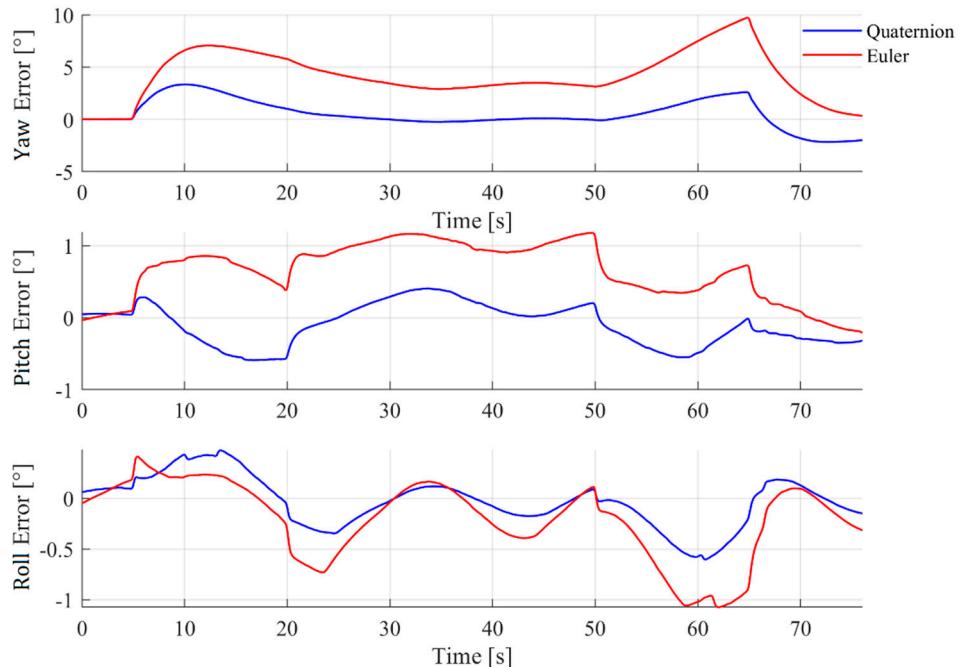
First, the turns at a 30-degree roll angle were simulated, with the results as shown in (Figures 7 and 8).



**Figure 7. Cont.**



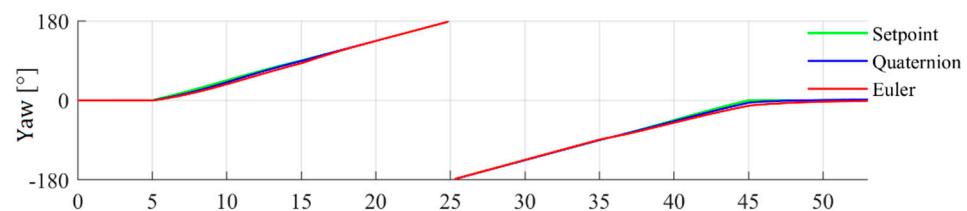
**Figure 7.** Setpoint tracking performance—turn at a 30-degree roll angle.



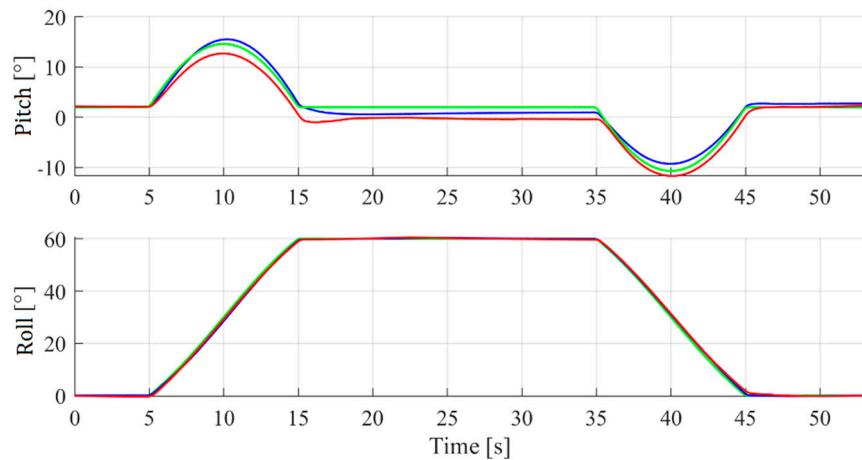
**Figure 8.** Tracking error—turn at a 30-degree roll angle.

At the 30-degree roll angle, both controllers perform the turn very well. The quaternion-based controller has a slight edge in tracking yaw and pitch angles, though.

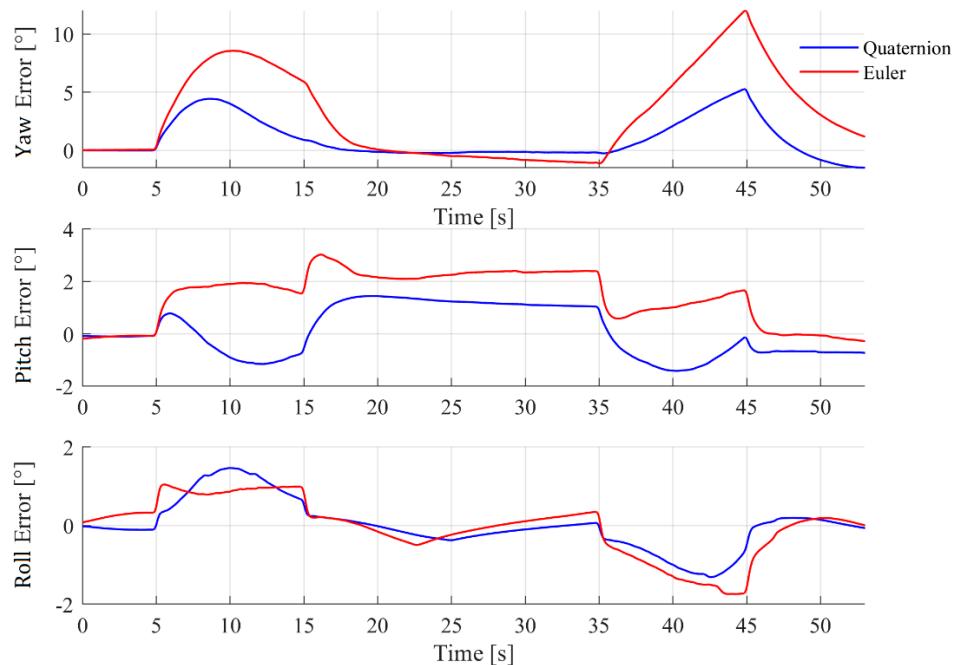
Next, the turns at a 60-degree roll angle were simulated, with the results as shown in (Figures 9 and 10).



**Figure 9. Cont.**



**Figure 9.** Setpoint tracking performance—turn at a 60-degree roll angle.

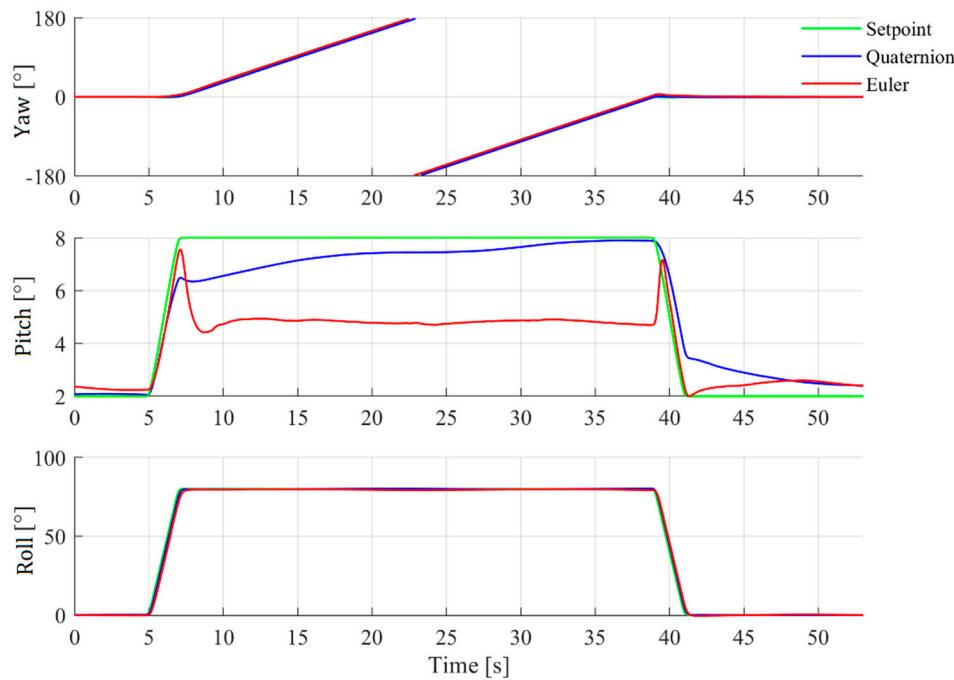


**Figure 10.** Tracking error—turn at a 60-degree roll angle.

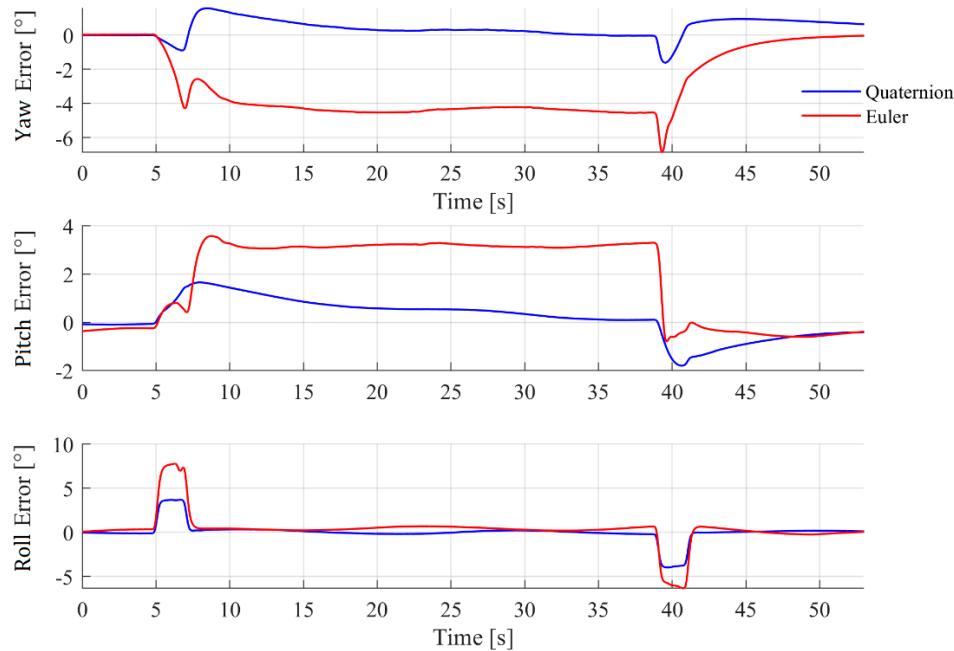
At the 60-degree roll angle, both controllers also perform well. Once again, the quaternion-based controller has a slight edge in tracking yaw and pitch angles.

Next, the turns at an 80-degree roll angle were simulated, with the results as shown in (Figures 11 and 12).

At the 80-degree roll angle, the difference in the performance of the two controllers becomes apparent. The quaternion-based controller performs significantly better with a much tighter tracking of yaw and pitch angles. The Euler-based controller performs worse than the quaternion one due to the couplings: rudder–pitch, and elevator–yaw in the aircraft’s control. The Euler-based controller does not “understand” that at high roll angles, the rudder controls the pitch angle, and the elevator controls the yaw angle. Due to its architecture and mathematical properties of quaternions, the quaternion-based controller “understands” this phenomenon and is able to perform proper and efficient control of the aircraft.



**Figure 11.** Setpoint tracking performance—turn at an 80-degree roll angle.

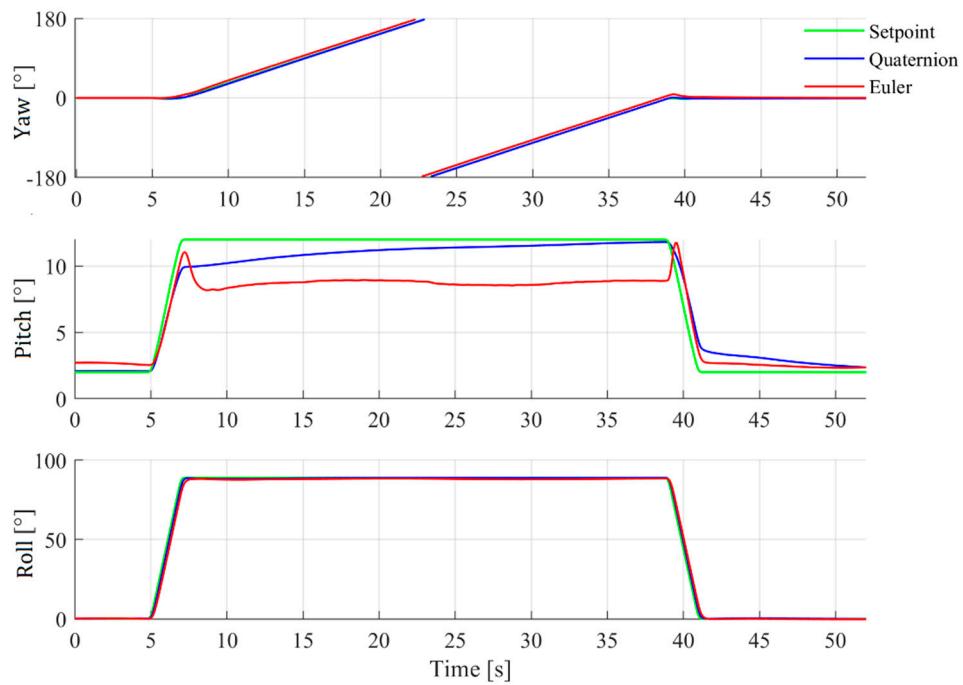


**Figure 12.** Tracking error—turn at an 80-degree roll angle.

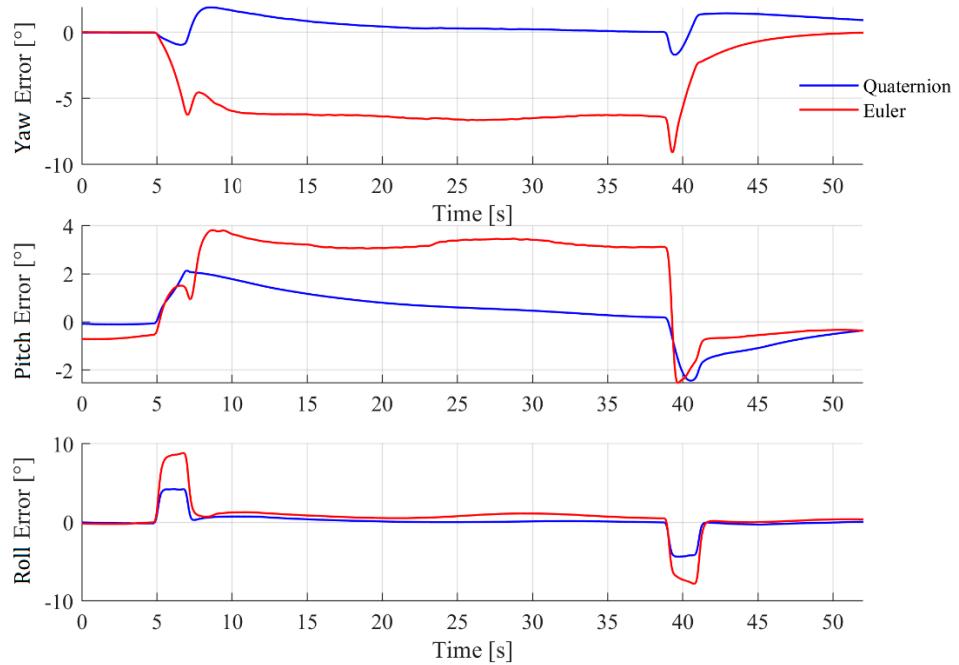
Lastly, the turns at a 90-degree roll angle were simulated, with the results as shown in (Figures 13 and 14).

At the 90-degree roll angle, the difference in the performance of the two controllers becomes even more significant. The quaternion-based controller continues to perform better and better as the roll angle increases, compared to the Euler-based controller. It is due to stronger rudder–pitch and elevator–yaw couplings as the roll angle increases.

Because the dynamic model was purchased and there was no insight into it, the experimental stability analysis of the controller was performed. The quaternion-based controller response was evaluated at various attitudes and various setpoints resulting in attitude errors spanning from 0 to 180 degrees in all three Euler angles. The experimental stability analysis did not show any unstable areas in the system.



**Figure 13.** Setpoint tracking performance—turn at a 90-degree roll angle.



**Figure 14.** Tracking error—turn at a 90-degree roll angle.

The proposed controller outperforms the Euler angle-based controller. The difference in performance between the two controllers becomes larger as the pitch and roll angles of the aircraft's maneuver increase. Where Euler-based attitude control is sufficient at low pitch and roll angles, it becomes lacking at high angles, mostly due to the elevator and rudder couplings. The major advantage of the quaternion approach reveals itself at high roll and pitch angles. The major difference in the setpoint tracking is observed in the yaw and pitch channels. There is no relevant difference in the setpoint tracking for the roll channel between the two controllers. This is because the roll is the last rotation in Euler angles and is equivalent to the aircraft's rotation around its  $x$ -axis.

#### 4. Conclusions

The controller can be successfully employed to control manned and unmanned aerial vehicles operating at high pitch and roll angles. In addition, a slightly modified version of the controller can be used to control cruising missiles and space rockets. Some preliminary tests were conducted in this direction, employing a Matlab simulation of an ILR-33 Amber sounding rocket, with great results.

Quaternion attitude control, counterintuitively, can be less complex than Euler angle control since a single quaternion represents a single rotation. In contrast, a single set of Euler angles represent three consecutive rotations, adding to the complexity of the control.

**Author Contributions:** Conceptualization, M.G.; methodology, M.G., M.K. and C.S.; software, M.G.; validation, M.G. and M.W.; formal analysis, M.K., C.S. and K.B.; investigation, M.G.; resources, M.G.; data curation, M.W.; writing—original draft preparation, M.G.; writing—review and editing, M.K., C.S. and A.Z.; visualization, M.G. and M.W.; supervision, M.K., C.S. and A.Z.; project administration, M.K., C.S. and A.Z.; funding acquisition, M.K., C.S. and A.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Wie, B.; Barba, P.M. *Quaternion Feedback for Spacecraft Large Angle Maneuvers*; Ford Aerospace & Communications Corporation: Palo Alto, CA, USA, 1984.
2. Markley, F.L.; Crassidis, J.L. *Fundamentals of Spacecraft Attitude Determination and Control*; Springer: Berlin/Heidelberg, Germany, 2019.
3. Chaturvedi, N.A.; Sanyal, A.K.; McClamroch, N.H. Rigid-Body Attitude Control. *IEEE Control. Syst.* **2011**, *31*, 30–51. [[CrossRef](#)]
4. Kuipers, J.B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, USA, 1999.
5. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35.
6. Graf, B. Quaternions and dynamics. *arXiv* **2008**, arXiv:0811.2889.
7. Ogata, K. *Modern Control Engineering*; Prentice Hall: Englewood Cliffs, NJ, USA, 2010.
8. Blajer, W.; Kosma, Z.; Krawczyk, M.; Graffstein, J. Prediction of the dynamic characteristics and control of aircraft in prescribed trajectory flight. *J. Theor. Appl. Mech.* **2001**, *39*, 79–103.
9. Blajer, W.; Graffstein, J.; Krawczyk, M. UAV program motion and control in prescribed mission. *J. Theor. Appl. Mech.* **1998**, *36*, 963–978.
10. Cook, M.V. *Flight Dynamics Principles*; Butterworth-Heinemann: Oxford, UK, 2013.
11. Krawczyk, M.; Szczepanski, C.; Zajdel, A. Simulation and Testing of Flight Stabilization System Using Trimmers. In *Automation 2021: Recent Achievements in Automation, Robotics and Measurement Techniques*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 185–196, ISBN 978-3-030-74893-7.