ASTEROID MISSION GUIDANCE AND CONTROL USING DUAL QUATERNIONS

MASTER THESIS

by

Marc Trullas Ballester

in partial fulfillment of the requirements for the degree of

Master in
Aeronautical Engineering - Space

at Universtitat Politècnica de Catalunya,
3rd of October 2019

- APPENDICES -

Tutor: Dr. ir. D. Garcia-Almiñana, UPC
Supervisors: Prof. dr. ir. J. McMahon, CU Boulder
Dr. ir. E. Mooij, TU Delft
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>Appendix A: Stability theory</td>
<td>1</td>
</tr>
<tr>
<td>Appendix B: Other system test results</td>
<td>5</td>
</tr>
<tr>
<td>Bibliography</td>
<td>17</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

B.1 Orbiting state variables and reference values with classic approach ....................... 6
B.2 Orbiting state variable errors with respect to reference trajectory with classic approach ...... 6
B.3 Orbiting control forces and torques commands with classic approach ........................ 7
B.4 Orbiting state variables and reference values with dual quaternion approach ................ 8
B.5 Orbiting state variable errors with respect to reference trajectory with dual quaternion approach 8
B.6 Orbiting control forces and torques commands with dual quaternion approach ............ 9
B.7 Hovering state variables and reference values with classic approach .......................... 10
B.8 Hovering state variable errors with respect to reference trajectory with classic approach .... 10
B.9 Hovering control forces and torques commands with classic approach ..................... 11
B.10 Hovering state variables and reference values with dual quaternion approach ............. 12
B.11 Hovering state variable errors with respect to reference trajectory with dual quaternion approach 12
B.12 Hovering control forces and torques commands with dual quaternion approach .......... 13
B.13 Landing state variables and reference values for the landing scenario with dual quaternion approach ................................................................. 13
B.14 Landing state variable errors with respect to reference trajectory with dual quaternion approach 14
B.15 Landing control forces and torques commands with dual quaternion approach ............ 14
APPENDIX A

STABILITY THEORY

To understand and design controllers capable of stabilizing spacecraft it is important to bear some theory and definitions in mind. This appendix gives the necessary theory to understand the design of the controllers proposed in Chapter 5. All this theory has been taken from [Schaub and Junkins, 2009].

STABILITY DEFINITIONS

There are different types of stability. Let \( x \) be a generalized state vector; then nonlinear dynamical systems can be written in the form

\[
\dot{x} = f(x, t)
\]  
(A.1)

To define stability of a dynamical system, some notions of an equilibrium state \( x_e \) and nominal reference motion \( x_r \) are required.

- **Equilibrium state**: A state vector point \( x_e \) is said to be an equilibrium state (or equilibrium point) of a dynamical system described by \( \dot{x} = f(x, t) \) at time \( t_0 \) if

\[
f(x_e, t) = 0 \quad \forall t > t_0
\]  
(A.2)

If the dynamical system is to follow a prescribed motion, then this motion is referred to as the nominal reference motion \( x_r(t) \). To describe the proximity of one state to another, the notion of neighborhoods is defined.

- **Neighborhood \( B_\delta \)**: Given \( \delta > 0 \), a state vector \( x(t) \) is said to be in the neighborhood \( B_\delta(x_r(t)) \) of the state \( x_r(t) \) if

\[
\|x(t) - x_r(t)\| < \delta \implies x(t) \in B_\delta(x_r(t))
\]  
(A.3)
A simple form of stability is the concept of a motion simply being bounded (or Lagrange stable) relative to $\mathbf{x}_r(t)$. Note that $\mathbf{x}(t_0)$ could lie arbitrarily close to $\mathbf{x}_r(t_0)$ while $\mathbf{x}(t)$ may still deviate from $\mathbf{x}_r(t)$. The only stability guarantee made here is that this state vector difference will remain within a finite bound $\delta$.

- **Lagrange stability**: The motion $\mathbf{x}(t)$ is said to be Lagrange stable (or bounded) relative to $\mathbf{x}_r(t)$ if there exists a $\delta > 0$ such that
  \[ \mathbf{x}(t) \in B_\delta(\mathbf{x}_r(t)) \]  
  (A.4)

Declaring a motion to be Lyapunov stable (also referred to simply as being stable) is a stronger statement than saying it is Lagrange stable.

- **Lyapunov stability**: The motion $\mathbf{x}(t)$ is said to be Lyapunov stable (or stable) relative to $\mathbf{x}_r(t)$ if for each $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that
  \[ \mathbf{x}(t_0) \in B_\delta(\mathbf{x}_r(t_0)) \implies \mathbf{x}(t) \in B_\delta(\mathbf{x}_r(t)) \quad \forall t > t_0 \]  
  (A.5)

In other words, if the state vector $\mathbf{x}(t)$ is to remain within any arbitrarily small neighborhood $B_\epsilon$ of $\mathbf{x}_r(t)$, then there exists a corresponding initial neighborhood $B_\delta(\mathbf{x}_r(t_0))$ from which all $\mathbf{x}(t)$ must originate. A stronger stability statement is to say the motion $\mathbf{x}(t)$ is asymptotically stable. In this case the difference between $\mathbf{x}(t)$ and $\mathbf{x}_r(t)$ will approach zero over time.

- **Asymptotic Stability**: The motion $\mathbf{x}(t)$ is asymptotically stable relative to $\mathbf{x}_r(t)$ if $\mathbf{x}(t)$ is Lyapunov stable and there exists a $\delta > 0$ such that
  \[ \mathbf{x}(t_0) \in B_\delta(\mathbf{x}_r(t_0)) \implies \lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}_r(t) \]  
  (A.6)

Except for the Lagrange stability definition, all other types of stabilities defined are referred to as local stability. The initial state vector has to be within a certain neighborhood $B_\delta$ relative to the desired state vector for stability to be guaranteed. If stability is guaranteed for any initial state vector $\mathbf{x}(t_0)$, then the system is said to be globally stable or stable at large.

**LYAPUNOV DIRECT METHOD**

Proving stability of nonlinear systems with the basic stability definitions and without resorting to local linear approximations can be quite tedious and difficult. Lyapunov’s direct method provides a tool to make rigorous, analytical stability claims of nonlinear systems by studying the behavior of a scalar, energy-like Lyapunov function. A major benefit of this method is that this can be done without having to solve the nonlinear differential equations.

To prove stability of a dynamical system, special positive definite functions called Lyapunov functions are sought.

- **Lyapunov function**: The scalar function $V(\mathbf{x})$ is a Lyapunov function for the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ if it is continuous and there exists a $\delta > 0$ such that for any $\mathbf{x} \in B_\delta(\mathbf{x}_r)$
1. \( V(x) \) is a positive definite function about \( x_r \).
2. \( V(x) \) has continuous partial derivatives.
3. \( \dot{V}(x) \) is negative semidefinite.

Even though \( V(x) \) explicitly depends on only the state vector \( x \), because \( x(t) \) is time varying, the Lyapunov function \( V \) is time varying, too. Using the chain rule, the derivative of \( V \) is found to be

\[
\dot{V} = \frac{\partial V^T}{\partial x} f(x) \tag{A.7}
\]

Bearing this definitions in mind, the following theorems can be stated:

- **Theorem A.1 - Lyapunov stability** If a Lyapunov function \( V(x) \) exists for the dynamical system \( \dot{x} = f(x) \), then this system is stable about the origin.

- **Theorem A.2 - Asymptotic stability** Assume \( V(x) \) is a Lyapunov function about \( x_r(t) \) for the dynamical system \( \dot{x} = f(x) \); then the system is asymptotically stable if
  1. the system is stable about \( x_r(t) \).
  2. \( \dot{V}(x) \) is negative definite about \( x_r(t) \).

And this is theorem that has been used to proof the stability of the controllers developed in this master thesis work.

**Barbalat’s theorem**

Sometimes, for time varying systems, it is difficult to find a Lyapunov function with a negative definite derivative. This is where Barbalat’s theorem comes into play. It says:

- **Barbalat’s lemma** if \( V(x, t) \) satisfies the following conditions:
  1. \( V(x, t) \) is lower bounded
  2. \( \dot{V}(x, t) \) is negative semi-definite
  3. \( \dot{V}(x, t) \) is uniformly continuous in time (satisfied if \( \ddot{V} \) is finite); then

\[
\dot{V}(x, t) \to 0 \quad \text{as} \quad t \to \infty \tag{A.8}
\]

This lemma is used during the proof of stability of the dual quaternion controller.
This appendix contains all the results from the system tests explained in Subsection 7.2.4. For the sake of readability, only the results from the landing scenario with classical representation were shown. Thus, this appendix shows all the remaining results to show that all system tests were successful.

**Orbiting - Classic**

To start with, results from the system test with the classical approach for the orbiting scenario are presented below. The order in which the results are presented follows the same structure as in the main document, (Subsection 7.2.4). First of all, the different state variables evolution over time is presented in Figure B.1. As can be seen, it corresponds to one orbital period. The upper-left plot corresponds to the position components, whereas the upper-right plot shows the linear velocities. Both of them start and finish at the same values due to the stable orbit around Bennu. On the bottom-left side, the quaternion orientation is presented. Then, on the bottom-right side of Figure B.1, the angular velocities are shown. The blue box corresponds to a zoom-in of a specific part of the plot for better appreciation of how variables converge to their reference values.

Next, Figure B.2 shows the errors of the attitude quaternion and the angular velocity with respect to the reference values. It is worth remembering that for the orbiting scenarios, only the rotational motion was controlled. For this reason, only the pointing errors and the angular velocity errors are presented. As can be seen, after a settling time, both motions reach stability and remain within the requirements specified. For the pointing error, there is no overshoot and the error goes directly to zero from its starting point. However, for the angular velocity, there is a clear overshoot before reaching stability. The settling time is about 400 s. The blue box is again the zoom-in area for better visualizing how each error is converging to its reference value.
Finally, the different control commands torques are presented in Figure B.3. It is interesting to notice that only the first 900 s are shown in this figure, for better appreciating how the controller manage to steer the spacecraft back to the reference trajectory. During the first seconds, the controller is clearly saturated reaching its maximum allowed values. Due to actuators limitations, there is a maximum torque of 0.2 Nm in each body direction that cannot be exceeded. During the first 300 s, the actuators are saturated before stabilizing the spacecraft to the reference trajectory. It is also worth mentioning that, after stabilization, the
B. OTHER SYSTEM TEST RESULTS

Figure B.3: Orbiting control forces and torques commands with classic approach

control torque commands are not zero. They have the values required to follow the reference trajectory. However, since these values are so small, they cannot be appreciated in this figure.

As a conclusion, the system test for the classical orbiting scenario is a success. All variables manage to get stabilized in the reference trajectory and remain there for the duration of the simulation. This proves that the simulator and the controller as a whole system works as expected.

ORBITING - DUAL QUATERNIONS

Next, results from the system test with the dual quaternion approach, for the orbiting scenario, are presented. They are very similar to the ones of the classical approach. First of all, the different state variables evolution over time is presented in Figure B.4. The upper-left plot corresponds to the position components, whereas the upper-right plot shows the linear velocities. Their values are identical to the previous system test with classical control. However, on the bottom-left side, the quaternion orientation plot shows some differences. More specifically, they look anti-symmetric with respect to the classical control. The reason for this is because the initial orientation error was set to be 180°, as it can be seen in Table 7.3. The classical control approach forced the spacecraft to rotate in one direction, whereas the dual quaternion controller chose the other one. This can also be observed in the bottom-right side of Figure B.4. The angular velocities of the body x and y axes are exactly the opposite than the ones with classical control. The blue box corresponds to a zoom-in of a specific part of the plot for better appreciation of how variables converge to their reference values.

Moreover, Figure B.5 shows the errors of the attitude quaternion and the angular velocity with respect to the reference values, for the dual quaternion orbiting system test. Again, only the pointing error and the angular velocity errors are shown because no orbit control has been applied. On the upper part, the pointing error goes from 180° to almost 0° very quickly (in about 500 s). The difference between the dual quaternion approach and the classical one is that the first one reduces the error much faster than the second one. The same can be noticed from the angular velocity errors shown on the bottom part of the figure. The overshoot for the angular velocities is as high as the classical one. The blue box is again the zoom-in area for better visualizing how each error is converging to its reference value.

Finally, the different control commands torques for the dual quaternion representation are presented in Figure B.6. Two aspects are worth mentioning when comparing these results with the classical ones. The first
one is that the settling time is shorter. The second one is that the control torque commands have the opposite shape than for the classical approach. This is the reason why the state variables look anti-symmetric. As has been said already, the dual quaternion approach chose to steer the spacecraft in the exact opposite direction than the classical control, when starting with 180° of pointing initial error. Furthermore, it is also important to notice that only the first 900 s have been represented in this figure to appreciate how the control commands evolve overtime. Otherwise, it would have been a straight line very close to zero, over the more than 7000 s of
This system test proves that the simulator developed with the implemented controller work as expected and manage to achieve the mission requirements, both with a classical representation and with the dual quaternion one. All sub-blocks defined in Section 7.1 work together as a system successfully.

**Hovering - Classic**

Next, results from the system test with the classical approach, for the hovering scenario, are presented below. As has been done in previous sections, the different state variables evolution over time are presented in Figure B.7. The upper-left plot corresponds to the position components. The reference values are also shown in dashed lined. Since the trajectory is an inertial hovering about Bennu, the position components are straight lines. The upper-right plot of the figure shows the velocity components. As can be seen, there is a clear overshoot to correct the position of the satellite and then, all velocity components go to almost zero. In this case, the overshoot has an order of magnitude of 0.1 m/s. It is important to remember that the initial conditions and gain settings for this simulation have been stated in Sub-section 7.2.4. The bottom-left side of Figure B.7 shows the different quaternion components over time, with the reference values. And last, in the bottom-right side of the figure, the angular velocity components are represented. They also show a clear overshoot, as the linear velocity did. In this case, the order of magnitude is about 0.08°/s. The reason for that is because the orientation of the satellite had to be corrected changing the angular velocity, to finally stabilize on the reference value.

Figure B.8 shows the errors of each state variable over time. As can be seen, the linear and angular velocities look exactly the same as in Figure B.7. This is because the reference velocities are zero. However, the a blue box have been inserted to zoom in and appreciate that both state variables actually converge within the requirements of the mission. Regarding the left part of the figure, the position and attitude errors are shown. They quickly go to zero (in about 40 s). It is important to mention that, in contrast with the orbiting scenario, orbit control was required for hovering. For this reason, position and velocity errors are presented in Figure B.8.

Finally, both the force and torque control commands are represented in Figure B.9. There is a clear saturation of the actuators in both cases before reaching stability. It is worth mentioning that only the first 55 s are
shown in this figure, for better appreciating how each controller manage to stabilize the spacecraft. As can be seen, it takes a little bit longer for the attitude motion to reach stability. Last, it is important to realize that the control commands never go to exactly zero values. There is always a tinny torque or force to be compensated so that inertial hovering can be achieved.

The conclusion for this system test is exactly the same than for the previous ones. It proves that the sim-
B. OTHER SYSTEM TEST RESULTS

Figure B.9: Hovering control forces and torques commands with classic approach

ulator developed and the classical control for hovering works as expected and provides us with valid results.

HOVERING - DUAL QUATERNIONS

Here, results from the system test with the dual quaternion approach, for the hovering scenario, are presented. Again, they are very similar to the ones of the classical approach. The different state variables evolution over time is presented in Figure B.10. The distributions of the different state variables is the same as in the previous cases. The main difference with respect to the classical approach is that the overshoot for the angular velocity is slightly smaller. In this case, the order of magnitude is $0.07^\circ/s$ instead of $0.08^\circ/s$. For the linear velocity, they remain equal.

Figure B.11 shows the different errors for each state variable. The performance of the dual quaternion controller is very similar to the classical one for this hovering scenario. The only difference is the already mentioned smaller overshoot for the angular velocity. The blue box areas are a zoom-in of specific regions of the plot to clarify how each state variable converges to the reference solution.

Finally, Figure B.12 shows both the force and torque commands for the dual quaternion controller in the hovering scenario. As can be seen, the actuators a saturated for almost 15 s and then manage to stabilize the spacecraft to the hovering trajectory. The settling time is this case is the same as for the classical representation.

LANDING - DUAL QUATERNIONS

Last but not least, results form the landing scenario with the dual quaternion approach are represented below. For the sake of simplicity, only the classical representation for the landing scenario was shown in the
main document. Here, system test results using the dual quaternion approach are shown to prove that everything implemented in this thesis, works properly and as expected. First, Figure B.13 shows the different state variables evolution over time. All simulation settings and initial conditions are described in Subsection 7.2.4 of the main document. The gains used for this simulations are also stated. As can be seen, all different state variables manage to follow the reference trajectory (shown in dashed lines) to finally land on the surface of
B. OTHER SYSTEM TEST RESULTS

Figure B.12: Hovering control forces and torques commands with dual quaternion approach

Figure B.13: Landing state variables and reference values for the landing scenario with dual quaternion approach

Bennu meeting the specified requirements.

Figure B.14 shows the different errors for each state variable. As can be appreciated, the initial conditions are not the nominal ones. For this reason, there is a clear overshoot in both the angular and linear velocities to compensate for these errors, and then follow the reference trajectory. The maximum overshoot for the linear velocity has an order of magnitude of 0.15 m/s, whereas it is 0.06°/s for the angular velocity. On the bottom-left side of the figure, the pointing error is represented showing how quickly the controller manages to correct the initial attitude error. Comparing these results with the ones in the main document for the
Figure B.14: Landing state variable errors with respect to reference trajectory with dual quaternion approach

classical controller, it can be said that the overshoot for the angular velocity here, is slightly smaller than the 0.08°/s for the classical control. However, for the linear velocity, it is a little bit bigger (compared to 0.1 m/s of the classical representation). The blue boxes still represent the zoom-in areas for better appreciating how each error converges to zero.

Figure B.15: Landing control forces and torques commands with dual quaternion approach

Finally, Figure B.15 shows both the force and torque commands of the dual quaternion controller used to guide the spacecraft through the landing trajectory. In this case, only for the first 15 s the actuators are saturated. Then, they converge to almost zero, where the spacecraft reaches convergence. It is important
noticing that they never go to exactly zero values, because there are always some tinny torques or forces to be performed so that the spacecraft follows the reference trajectory.

The final conclusion of this Appendix B is that all system tests carried out have been successful. The different blocks defined in Section 7.1, can work together as a system and provide the user with interesting and valid results, which can be used to answer the research question of this master thesis. For all different scenarios defined in this work, both the classical and the dual quaternion controllers manage to steer the spacecraft through the reference trajectory meeting the mission requirements. The purpose of this Appendix is to prove and show the system test results of all the different simulations that have been carried out in this thesis, and that could not be shown in the main document for the sake of simplicity.