Agile hopping robot for lunar exploration targeting data gathering and sample collection

João Francisco Formigão e Santos
joao.f.formigao@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2019

Abstract

The growing number of space missions has led to the application of the concept of hopping robots for autonomous locomotion in planetary and small-body environments. The strong jumping ability of this type of robots allow them to move in such complex environments with high efficiency.

The main challenge to this concept has been related to impacts with the ground. To solve this issue, this thesis explores the application of a Double Gimbal Variable Speed Control Moment Gyroscope to control the attitude of a hopping robot during a ballistic flight. During flight, this actuator must be able to guarantee an attitude and angular velocity of the robot such that, when the robot hits the ground, it has a safe position. Additionally, in flight, the robot must be able to perform observation maneuvers that require the robot to achieve a certain attitude with zero angular velocity.

To achieve these objectives, this thesis presents the system equations of motion. Then, a broad study of the controllability is performed. Finally, three control architectures are designed and tested. Among these a strategy using Trajectory Optimization is constructed and applied. The final proposed controller is a closed-loop controller following these optimized trajectories, combining efficiency with robustness. The results are presented and compared with the previous solutions reviling an advantage in the use of this architecture.

Keywords: hopping robots, attitude control, Double Gimbal Variable Speed Control moment gyroscope, trajectory optimization

1. Introduction

With the recent trend toward more frequent space missions to Mars and other celestial bodies such as moons asteroids and comets [3], the field of hopping robotics achieved a new application. These celestial bodies are usually characterized by a low to medium gravitational environment and unstructured terrain. For exploration in such complex scenarios, an interest in multi-functional vehicles, capable of providing high mobility for scientific packages has sparked the interest of the space exploration community in this field [3].

According to [7], for the design of a hopping robot, the jumping process is divided into three stages: takeoff, air posture adjustment, and landing. This segmentation of the process is useful for control and actuation purposes. By separating into three stages the control can be applied separated from each phase.

The takeoff stage starts when the robot is in equilibrium on the ground and ends when the foot takes off the ground towards the air [7]. It is during this phase that occurs the conversion of potential energy(stored in the actuator) into kinetic energy. Regarding the take off mechanism, hoppin robots can be described as [7]: reaction-propulsion drive (propelled as result of chemical reaction), pneumatic drive (the hop results from the contraction of series of pneumatic structures to accomplish), spring drive (vertical hopping motions are achieved by the release of a spring system) and flexible material drive (take off mechanism takes advantage of elastic properties of certain materials such as fiberglass or shape memory alloy).

The second stage of the cycle corresponds to the air posture adjustment during the flight phase. In this phase, the robot adjusts its orientation relative to the ground in order to achieve a safe landing. Some bio inspired robots make use of wings and artificial tails. Other robots use attitude actuator mechanisms originally designed for spacecraft such as reaction-propulsive systems or momentum exchange devices. Simpler robots even present no attitude actuation and achieve a desired position after landing and by means of its external structure.

Finally, there is the landing stage. This phase starts when the robot touches the ground and finishes when it reaches a static equilibrium. This phase is important for protecting the structure from the rigid collision with the ground. For some robots it is achieved by a real-time anticipatory control over the take off actuators that receive the first impact with the ground. When this is not possible, usually the focus is on the development of an independent buffering mechanism.

1.1. Moonhopper

The Moonhopper is a spring drive hopping robot with a Double Gimbal Variable Speed Control Moment Gyroscope (DGVSCMG) as a air posture actuator and with no landing buffer. Figure 1 presents an image of the Moonhopper. Dividing into four parts, the robot is composed by: DGVSCMG (the attitude control actuator); Robot body (support for the DGVSCMG, for the electric components,the batteries and the motor responsible for taking off); Robot legs (geared six-bar spring/linkage system responsible for take off) and Robot foot (circle shaped it supports all the other components).

In order to perform a jump, first, the robot contracts its legs by means of a wire that connects the robot body with the foot. By using a system of gears connected to a rotor, the wire is retracted and the legs are compressed. As a consequence the springs are extended, storing the potential energy for the jump. When the legs are fully compressed (and the springs fully extended), a fixed gear locks the releasing mechanism. To jump, this locking mechanism is released and the springs contract. This extension of the legs creates a force on the ground that is responsible for inducting an initial velocity to the robot ballistic trajectory.

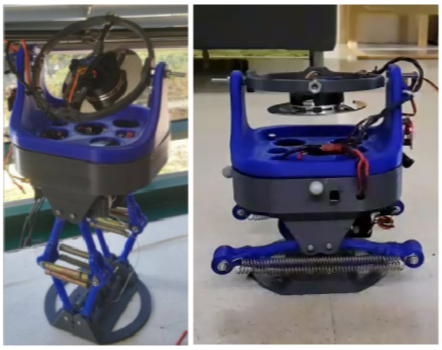

Figure 1: The Moonhopper. Extended and compressed configuration

The robot lacks a take off direction mechanism and a self righting mechanism, features commonly found in hopping robots designed for extraterrestrial exploration. For this reason, the prototype can only accomplish vertical jumps.

1.2. Related Work

The design of the Moonhopper was inspired by the work of J. Burdick and P. Fiorini in their works [3] and [6]. In [3], the authors present three generation designs for hooping robots projected for lunar exploration. The Moonhoper structure and take off mechanism is closely related to the second generation robot presented in [3]. More precisely, the six bar spring/linkage structure is copied from this second generation. The mechanism presents force/displacement profile (see [3]) that reduces the likelihood of premature lift-off due to the shocks inherent in initial spring release. Additionally, the motor peak design torque is lowered as compared with a linear spring leg robot. These features are the reason for the high efficiency (η) of the second generation.

Regarding the attitude actuation mechanism, the DGVSCMG choice as a momentum exchange device results from the work presented in [6]. In this article, the proposed actuation consists of two reaction wheels aligned with the y and z axes of the body. These actuators are supposed to provide the system with full attitude control in the case of no initial angular velocity. When the ballistic trajectory starts with initial angular velocity it is possible only to achieve control over the direction of the axis that is perpendicular to the axes of the two reaction wheels.

Instead of using reaction wheel, this research introduces the DGVSCMG as air posture actuator. This device consists of a spinning wheel with its axis controlled by two gimbals. In addition, the velocity of this wheel can be controlled. In general, a single DGVSCMG is considered as a method for three-dimensional attitude control as it is able to introduce torques in three directions.

In this work, the attitude dynamics of the Moonhopper are developed and presented in a state space formulation. Using a model of the Moonhopper developed in Gazebo/ROS, the results from the simulation of the state space model are compared with the Gazebo model for validation. With the description of the attitude dynamics, two attitude controllers are adapted from spacecraft attitude control and applied to the robot model. In addition, a Trajectory Optimization formulation is constructed. The goal is to obtain optimized attitude trajectories to be followed by the system and compare the results with a classical approach. The final controller is expected to combine these two strategies: trajectory optimization for trajectory planning and a closed loop system that follows the trajectories.

2. State Space System

The model developed in this research is a modification of the model developed in [5] of the attitude of an DGVSCMG actuated satellite. The difference is found in the position and configuration of the DGVSCMG. In this configuration, this device is placed away from the robot center of mass (CoM). Also, the wheel is placed off centered from the gimbals axes (see Figure 1).

The description of the dynamics uses a specific notation that requires description. In Figure 2 a representation of the top view of the DGVSCMG is presented. Frame F defines the orientation of the DGVSCMG in the robot body but stays fixed with respect to the body. In figure 2, $\hat{f}_1$, $\hat{f}_2$, and $\hat{f}_3$ are the orthogonal unit vectors of frame F. Frame G stays fixed with the outer gimbal and rotates with respect to frame F an angle ψ along axis $\hat{f}_1$. Frame G s defined by the orthogonal vectors $\hat{g}_1$, $\hat{g}_2$, and $\hat{g}_3$. Finally, frame H stays fixed with the inner gimbal and rotates with respect to frame G an angle θ along axis $\hat{g}_2$. Using $[FG]$ as the rotation matrix of frame G with respect to frame F, this matrix is

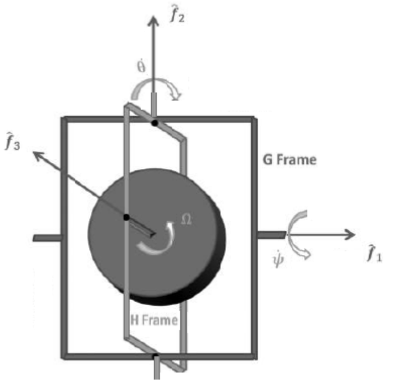

Figure 2: Representation of the DGVSCMG (top view)

defined as the cosine matrix of the rotation of ψ on axis $\hat{f}_1$. $[GH]$, following the notation, is the cosine matrix of the rotation of θ on axis $\hat{g}_2$.

The rotation of each frame with respect to the inertial frame defined outside the robot (frame N), can be written in terms of the unit vectors $\hat{f}_1$, $\hat{g}_2$, $\hat{h}_3$ and the angular rates $\dot{\psi}$, $\dot{\theta}$ and Ω: $\omega_{F/N} = \omega$; $\omega_{G/N} = \omega + \omega_{G/F} = \omega + \dot{\psi}\hat{f}_1$; $\omega_{H/N} = \omega + \dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2$ and $\omega_{W/N} = \omega + \dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2 + \Omega\hat{h}_3$.

These relations are used to solve for the time derivative of the angular momentum H_i of each robot/device component i (robot body LBS, the G and H structure frames, and the spinning wheel W). Each component angular momentum is defined using the formula for an off centered rotation of a rigid body, equation (1).

$$\begin{aligned} H_i &= [I_i]\omega_{i/N} + r \times m_i.v_{i/CoM} \\ &= [I_i]\omega_{i/N} - m_i\tilde{r}_i\tilde{r}_i\omega_{i/CoM} \end{aligned} \tag{1}$$

From (1), the Euler's rotational EOM can be applied to the combined angular momentum $H = H_{LBS} + H_G + H_H + H_W$. For the Moonhopper body, the angular momentum time derivative (in reference N) is given by the expression

$$\begin{aligned} H_{LBS} &=^F [I_{LBS}]\omega \\ \dot{H}_{LBS} &=^F [I_{LBS}]\dot{\omega} + \omega \times \left({}^F[I_{LBS}]\omega\right) \end{aligned} \tag{2}$$

Matrix ${}^F[I_{LBS}]$ corresponds to the combined inertia of the legs, foot and robot body. For the other components, the derivation requires some more attention. For G structure frame (outer gimbal),

$$\begin{aligned} H_G &= H_{G_c} + H_{G_{off}} \\ &=^F [I_G](\omega + \dot{\psi}\hat{f}_1) - m_G\tilde{r_G}\tilde{r_G}\omega \end{aligned} \tag{3}$$

$$\begin{aligned} &\dot{H}_{G_c} = \tfrac{{}^G d}{dt}\left({}^F[I_G]\,\omega_{G/N}\right) + \omega_{G/N} \times \left({}^F[I_G]\omega_{G/N}\right) \\ &=^F [I_G]\left(\tfrac{{}^G d}{dt}\left(\omega + \dot{\psi}\hat{f}_1\right)\right) + \omega_{G/N} \times \left({}^F[I_G]\omega_{G/N}\right) \\ &=^F [I_G]\left(\dot{\omega} + \ddot{\psi}\hat{f}_1 + \omega \times \left(\dot{\psi}\hat{f}_1\right)\right) \\ &\omega_{G/N} \times \left({}^F[I_G]\,\omega_{G/N}\right) \end{aligned} \tag{4}$$

By several iterations of the transport theorem ${}^F d({}^A v)/dt =^A d({}^A v)/dt + \omega_{A/F} \times^A v$ it is possible to remove $[I_G]$ from the derivative in the second line because it is constant in frame G.

$$\begin{aligned} \dot{H}_{G_{off}} &= {}^N\frac{d}{dt}\left(-m_G\tilde{r}_G\tilde{r}_G\omega\right) \\ &= -m_G\tilde{r}_G\tilde{r}_G\dot{\omega} + \omega \times \left(-m_G\tilde{r}_G\tilde{r}_G\omega\right) \end{aligned} \tag{5}$$

The result for $H_H =^F [I_H](\omega + \dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2) - m_H\tilde{r_H}\tilde{r_H}\omega$ follows the same procedure as for H_G.

$$\begin{aligned} &\dot{H}_H =^F [I_H]\left(\dot{\omega} + \ddot{\psi}\hat{f}_1 + \ddot{\theta}\hat{g}_2 + \omega \times \left(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2\right)\right. \\ &\left. + \left(\dot{\psi}\hat{f}_1\right) \times \left(\dot{\theta}\hat{g}_2\right)\right) + \omega_{H/N} \times \left({}^F[I_H]\,\omega_{H/N}\right) \\ &-m_H\tilde{r}_H\tilde{r}_H\dot{\omega} + \omega \times \left(-m_H\tilde{r}_H\tilde{r}_H\omega\right) \end{aligned} \tag{6}$$

For the reaction wheel H_W, the procedure is slightly different. The velocity of the wheel CoM depends on the rotation of the body and the rotation of the two gimbal axes. Which results in the expression: $H_W = H_{W_c} + H_{W_{off}}$ with $H_{W_c} = {}^F[I_W]\,\omega_{W/N} = {}^F[I_W]\left(\omega + \dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2 + \Omega\hat{h}_3\right)$ and $H_{W_{off}} = r_W \times (m_W(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2) \times r_{W/H}) + r_W \times (m_W\omega \times r_W)$. $r_{W/H} = [FH][0\,0\,r_{w_2}]^T$ is the displacement vector of the wheel CoM with respect to the gimbals CoM. Applying the procedures described above for the H_W

$$\begin{aligned} &\dot{H}_{W_c} =^F [I_W]\left(\dot{\omega} + \ddot{\psi}\hat{f}_1 + \ddot{\theta}\hat{g}_2 + \dot{\Omega}\hat{h}_3 + \omega \times \left(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2\right.\right. \\ &\left.\left. +\Omega\hat{h}_3\right) + \left(\dot{\psi}\hat{f}_1\right) \times \left(\dot{\theta}\hat{g}_2 + \Omega\hat{h}_3\right) + \left(\dot{\theta}\hat{g}_2\right) \times \left(\Omega\hat{h}_3\right)\right) \\ &+ \omega_{W/N} \times \left({}^F[I_W]\,\omega_{W/N}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{H}_{W_{off}} = D_r^F &\times \left(m_W\left(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2\right) \times r_{W/H} + \omega \times r_W\right) \\ &+ r_W \times \left(m_W\left(\ddot{\psi}\hat{f}_1 + \ddot{\theta}\hat{g}_2 + \omega_{G/F} \times \left(\dot{\theta}\hat{g}_2\right)\right) \times r_{W/H}\right. \\ &+m_w\dot{\omega} \times r_w + m_w\left(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2 + \omega\right) \times D_r^F\Big) \\ &+ \omega \times \Big(r_W \times (m_W(\dot{\psi}\hat{f}_1 + \dot{\theta}\hat{g}_2) \times r_{W/H}) \\ &+r_W \times (m_W\omega \times r_W)) \end{aligned} \tag{8}$$

Where D_r^F is the time derivative of r_w in frame F. The objective from this point is to obtain an expression for $\dot{\omega}$. Introducing the modified angular momentum derivatives $\dot{H}_i' = \dot{H}_i - [I_i]\dot{\omega} - (-m_i\tilde{r}_i\tilde{r}_i\dot{\omega} - \omega \times (m_i\tilde{r}_i\tilde{r}_i\omega))$ and combined moments of inertia $[I_{stat}] =^F [I_{LBS}] - m_G\tilde{r}_g\tilde{r}_g - m_h\tilde{r}_h\tilde{r}_h - m_w\tilde{r}_w\tilde{r}_w$ and $[I] = [I_{stat}] + {}^F[I_G] + {}^F[I_H] + {}^F[I_W]$. The expression of $\dot{\omega}$ is given by

$$\dot{\omega} = [I]^{-1}\left(-\omega \times [I_{stat}]\omega - \dot{H}_G' - \dot{H}_H' - \dot{H}_W'\right) \tag{9}$$

Since the DGVSCMG is commanded by torques, an additional relation regarding the inputs and the states of the system must be added. The goal of this second derivation is to obtain an equation of $\ddot{\psi}$, $\ddot{\theta}$ and $\dot{\Omega}$ as function of the states and input torques. For this purpose, the relations between the time derivatives of the angular momentum of the parts of the CMG and the respective input torques are introduced. Following [5] the relation is given by

$$[HF]\ {}^{F}\left(\dot{H}_{W_c}\right) = \begin{bmatrix} \sim \\ \sim \\ u_W \end{bmatrix} \tag{10}$$

$$[GF]\ {}^{F}\left(\dot{H}_{W_c} + \dot{H}_{W_{off2/H}} + \dot{H}_{H_c}\right) = \begin{bmatrix} \sim \\ u_H \\ \sim \end{bmatrix}$$

$${}^{F}\left(\dot{H}_{W_c} + \dot{H}_{W_{off1/H}} + \dot{H}_{W_{off2/H}} + \dot{H}_{H_c} + \dot{H}_{G_c}\right) = \begin{bmatrix} u_G \\ \sim \\ \sim \end{bmatrix}$$

From here it is possible to organize the previous equation and obtain an expression in the form $M[\ddot{\psi},\, \ddot{\theta},\, \dot{\Omega}]^T + B = [u_G,\, u_H,\, u_W]^T$. Inverting this expression will give the desired dynamics.

The state space formulation comprises the variables q, ω, ψ, θ, Ω, $\dot{\psi}$ and $\dot{\theta}$. The attitude kinematics are described using the quaternions formulation. The expression for the quaternion time derivative is given by equation (11)

$$\dot{q} = \frac{1}{2} \begin{bmatrix} q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \\ -q_x & -q_y & -q_z \end{bmatrix} \omega \tag{11}$$

3. Controllability

In this section, a broad study over the controllability and singularities is provided. The objective is to prove if given an initial state, the system is able to reach any desired attitude with a specific angular velocity ($SO(3) \times \mathbb{R}^3$), as it is done in [2].

Using the system described in the previous section and study its controllability characteristics requires numerical approximation and several assumptions as it is done in [5]. Instead, in this derivation, the system that is used is a simplified version. Assuming $H \approx H_{stat} + H_{CMG}$, where $H_{stat} = {}^{F}[I_{stat}]\omega$ and $H_{CMG} = I_{wzz}\Omega \begin{bmatrix} sin(\theta) & -sin(\psi)cos(\theta) & cos(\theta)cos(\psi) \end{bmatrix}^T$.

Applying the time derivative to the expression results in: $[I_{stat}]\dot{\omega} + \dot{H}_{CMG} + \omega \times ([I_{stat}]\omega + H_{CMG}) = 0$. The expression of $\dot{H}_{CMG}$ is obtained using the procedure followed earlier for the complete system. The result is given by

$$\begin{aligned} \dot{H}_{CMG} &= \frac{{}^{F}d}{dt}\left(I_{w_{zz}}\Omega\hat{h}_3\right) = \frac{{}^{H}d}{dt}\left(I_{w_{zz}}\Omega\hat{h}_3\right) \\ &+ \omega_{H/F} \times \left(I_{w_{zz}}\Omega\hat{h}_3\right) \\ &= \begin{bmatrix} 0 & I_{w_{zz}}\Omega c_\theta & I_{w_{zz}} s_\theta \\ -I_{w_{zz}}\Omega c_\theta c_\psi & I_{w_{zz}}\Omega s_\theta s_\psi & -I_{w_{zz}} s_\psi c_\theta \\ -I_{w_{zz}}\Omega c_\theta s_\psi & -I_{w_{zz}}\Omega s_\theta c_\psi & I_{w_{zz}} c_\theta c_\psi \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\Omega} \end{bmatrix} \end{aligned} \tag{12}$$

Describing the attitude kinematics using a matrix $R \in SO(3)$ that represents the attitude of the the body with respect to a reference inertial frame, the kinematics are described by the equation $\dot{R}(t) = R(t)\mathbf{S}(\omega(t))$, where ω is the instantaneous body-frame components of the angular velocity of the body $\omega \in \mathbb{R}^3$ and $\mathbf{S}()$ represents the cross product operation. As stated in [2], on the sub manifold with constant angular momentum M_H the attitude kinematics equation can be written in terms of the conservation of angular momentum ($R^T H = I_{stat}\omega + H_{cmg}(\psi, \theta, \Omega)$) as

$$\dot{R}(t) = R(t)\mathbf{S}(I_{stat}^{-1}\{R^T(t)H - H_{cmg}(\psi, \theta, \Omega)\}) \tag{13}$$

Since (13) combines the kinematics and dynamics equation of (12), the system is fully defined on M_H by including $\dot{\psi} = u_1,\ \dot{\theta} = u_2,\ \dot{\Omega} = u_3$ to equation (13). The resulting equation define a control afine system of the form: $\dot{y}(t) = f_H(y(t)) + g_1(y(t))u_1(t) + g_2(y(t))u_2(t) + g_3(y(t))u_3(t)$. These dynamics are defined on the manifold $\mathbf{N}$, where $y = (R, \psi, \theta, \Omega) \in \mathbf{N}$ represents the body attitude and the DGVSCMG configuration. In this system

$$\begin{aligned} &f_H(R, \psi, \theta, \Omega) = \left(R\mathbf{S}(I_{stat}^{-1}\{R^T H - H_{cmg}(\psi, \theta, \Omega)\}), 0\right) \\ &g_1(R, \psi, \theta, \Omega) = (0, e_1) \\ &g_2(R, \psi, \theta, \Omega) = (0, e_2) \\ &g_3(R, \psi, \theta, \Omega) = (0, e_3) \end{aligned} \tag{14}$$

where $e_i \in \mathbb{R}^3$ is a vector whose i-th element is 1 and the remaining 0. The resulting system is relatively similar to the one from [2] regarding a spacecraft controlled by three single gimbal control moment gyros. In order to prove the controllability of system (14) on the respective manifold, two results must be proven: (1) the vector field f_H is weakly positively Poisson stable on $\mathbf{N}$; (2) the system (14) is strongly accessible and controllable on $\mathbf{N}$

For the first result, the proof presented in [2] can be directly applied to this case as it does not depend on the function $v(\theta)$, in this case $H_{cmg}(\psi, \theta, \Omega)$. For the second result, the result still applies, but is rather different. Let $H \in \mathbb{R}^3$. Let $\xi_{11} = [g_1, f_H]$, $\xi_{12} = [g_1, \xi_1]$ and $\xi_{13} = [\xi_1, \xi_2]$ be defined as the Lie Brackets and treating f_H and g_i as vector fields on $\mathbb{R}^{3x3} \times \mathbb{R}^3$. Using the expression for the Lie Brackets

$$\begin{aligned}&\frac{d}{dh}|_{h=0}\Big[\hat{f_H}((R,\psi,\theta,\Omega)+hg_1(R,\psi,\theta,\Omega))\\&-\hat{g}_1((R,\psi,\theta,\Omega)+hf_H(R,\psi,\theta,\Omega))]\end{aligned}\tag{15}$$

of [2]s, the resulting expression of ξ_{11}, ξ_{12} and ξ_{13} is given by

$$\begin{aligned}\xi_{11}&=-\big(R(t)\mathbf{S}(I_{stat}^{-1}\partial H_{cmg}/\partial\psi(\psi,\theta,\Omega)),0,0,0\big)\\\xi_{12}&=\big(R(t)\mathbf{S}(I_{stat}^{-1}\partial^2 H_{cmg}/\partial\psi^2(\psi,\theta,\Omega)),0,0,0\big)\\\xi_{13}&=\big(R\mathbf{S}(I_{stat}^{-1}\partial^2 H_{cmg}/\partial\psi^2(\psi,\theta,\Omega)\times\\&\quad I_{stat}^{-1}\partial H_{cmg}/\partial\psi(\psi,\theta,\Omega)),0,0,0\big)\end{aligned}\tag{16}$$

An analogous result can be obtained for the Lie Brackets regarding g_2 with the partial derivative with respect to θ instead of ψ. In this case only ξ_{13} is defined as $\xi_{13}=[g_3,f_H]$ since $\xi_{23}=0$ and $\xi_{33}=0$. ξ_{13} follows the same structure as for ξ_{11}.

As in [2], ξ_{i1} and ξ_{i2} are linearly independent as the vectors $\partial H_{cmg}/\partial\psi$ and $\partial^2 H_{cmg}/\partial\psi^2$ are perpendicular. ξ_{i3} is linearly independent as it is also perpendicular to the others. All Lie Brackets and functions g_i are linear independent between each other and contained in the strong accessibility algebra [2]. As result, the system (14) is strongly accessible. Strong accessibility implies accessibility, and assuming the weak positive Poisson stability of (14), the system is controllable in $\mathbf{M}_H$ following [2].

The result shows that system can be steered between any two states lying on the same constant angular momentum sub manifold by using suitable gimbal motions.

3.1. Singularities

This situation is well presented when working with double gimbal control moment gyros. It occurs when the outer gimbal axis is collinear with the reaction wheel axis [5]. In this case, an acceleration in the wheel produces the same effect as the acceleration in the outer gimbal axis, leading to a lost of one degree of freedom. When θ equals $\frac{\pi}{2}$ or $-\frac{\pi}{2}$, matrix (12) degenerates and looses one rank, resulting in

$$\begin{bmatrix}0 & 0 & I_{w_{zz}}\\0 & I_{w_{zz}}\Omega sin(\psi) & 0\\0 & -I_{w_{zz}}\Omega cos(\psi) & 0\end{bmatrix}$$

Finally, it is important to mentioned that the controllability of the system is not compromised. In [2],the authors prove that, for a system like this, the controllability in M_H exists despite the existence of singularities.

4. Controller

Using the system dynamics described in section 2, the goal is to control the attitude of the DGVSCMG actuated hopping robot. Experiments performed with the Moonhopper show that on the moon's surface, the robot is expected to jump and reach a maximum height of $5.69m$ during a time interval of $5.3s$. During this time the robot must be able to achieve one of two objectives: (1) from an initial state (orientation and angular velocity), achieve a landing position (defined later); (2) from an initial state, achieve a desired orientation (with 0 angular velocity) during 1 second, at the maximum height, and then return to a landing position.

In order to achieve these objectives three strategies were approached: (1) a classical attitude hold as a PD + Dynamic inversion; (2) an open loop system controlled by an input set obtained from a Trajectory Optimization method; (3) a nonlinear control analysis of a DGVSCMG.

The classical attitude hold approach is divided into two steps. The first step is to compute the required torque to be produced by the actuator to follow a specific trajectory. The time derivative of the angular momentum is divided into $\dot{H}_{LBS}+\dot{H}_{CMG}=0 \iff \dot{H}_{LBS}=-\dot{H}_{CMG}$. This can be read as: the torque to be applied to the robot to perform the maneuver is equal to the torque produced by the actuator. To compute the required torque a PD controller is used: $\dot{H}_{CMG}=K_P e_q+K_D e_\omega$, where e_q corresponds to the vector part of the error quaternion and $e_\omega=\omega_d-\omega$. The second step is to write the equation for $\dot{H}_{CMG}=\dot{H}_G+\dot{H}_H+\dot{H}_W$ in order to have an expression in the form of $\dot{H}_{CMG}=M_{\dot{H}_{CMG}}\begin{bmatrix}\ddot{\psi} & \ddot{\theta} & \dot{\Omega}\end{bmatrix}^T+B_{\dot{H}_{CMG}}$. Inverting the expression gives the requires values for $\ddot{\psi}$, $\ddot{\theta}$ and $\dot{\Omega}$. To compute the input torques, the procedure follows the relations in 10.

The trajectory optimization planned controller, makes use of input sets computed following a trajectory optimization method to be applied directly to the open loop system described in section 2. Trajectory Optimization methods are explored in the next section.

The nonlinear control analysis of a DGVSCMG is presented in [5]. Defining a Lyapunov function for the control dynamics $V(\delta\omega,\sigma)=\frac{1}{2}\delta\omega^T[I]\delta\omega+2K\ln\left(1+\sigma^T\sigma\right)$ the authors state the conditions in which this function is valid:

$$-[I]\dot{\omega}-\frac{1}{2}^B\frac{\mathrm{d}}{\mathrm{d}t}[I]\delta\omega=K\sigma+[P]\delta\omega-[I]\left(\dot{\omega}_r-\omega\times\omega_r\right)\tag{17}$$

this control law follows the nomenclature presented with the exception of σ, the modified rodrigues parameters (MRPs) of the current orientation relative to the desired orientation, and $\delta\omega=\omega_r-\omega$. K and $[P]$ are the gains.

The Lyapunov control law, departs this relation which is a function of the reference trajectory, the states and the inputs (for this case $\ddot{\psi}$, $\ddot{\theta}$ and $\dot{\Omega}$) and establishes conditions of validity. The conditions of validity in this case are stability constraints that, when solved result in input signals. Solving the stability constraint of 17 as in [5] and adapted to the system in 2, the result requires to solve an nonlinear matrix equation that cannot be solved analytically for the input vector. Instead, a Newton–Raphson (N–R) iteration is used to find the root of the relation (see [5]).

5. Trajectory Optimization

The term trajectory optimization refers to a set of methods that are used to find the best choice of trajectory, typically by selecting the inputs to the system, known as controls, as functions of time [4]. A trajectory optimization problem seeks to find the a trajectory for some dynamical system that satisfies some set of constraints while minimizing some cost functional. In order to construct a trajectory optimization problem it is required to specify: the system dynamics, boundary conditions (state, input and time), (nonlinear) constraints (state and input) and a cost functional. The trajectory optimization problem is then set as

$$\min_{u(t)} J(x_0, u_0, t_0, x_F, u_F, t_F) + \int_0^T w(x, u, t) d\tau \qquad (18)$$

$$\begin{aligned}
&Variables = \{\boldsymbol{x}(t),\ \boldsymbol{u}(t)\} \\
&Dynamics : \dot{x} = f(x, u) \\
&Constraints : g_{min} \leq g(t, x, u) \leq g_{max} \\
&Boundaries : \\
&\qquad \Psi(t_0, x_0, u_0) = \Psi_0 \\
&\qquad \Psi_{Fmin} \leq \Psi(t_F, x_F, u_F) \leq \Psi_{Fmax}
\end{aligned}$$

where x is the state vector and u is the input vector.

Solving the trajectory optimization problem can be accomplished by two types of methods: indirect or direct methods. Indirect methods analytically construct the necessary and sufficient conditions for optimality then discretize these conditions and solve them numerically [1]. Direct methods discretize the trajectory optimization problem itself, typically converting the original trajectory optimization problem into a nonlinear program. In general, direct methods are much easier to set up and solve. At the same time, they do not have a built-in accuracy metric and are less accurate since the solution can include discretization errors[1]. For these reasons, in this work, only direct methods are applied.

The direct method that is used is the trapezoidal direct collocation method. First, the trajectory is discretized in a finite number of points (N_{grid}) that represent a finite number of decision variables. At each point, a collocation constraint is imposed that relates the previous and the next points. This collocation constraint imposes the system dynamics to the resulting optimization problem. It follows from $\int_{t_k}^{t_{k+1}} \dot{x} dt = \int_{t_k}^{t_{k+1}} f(x, u) dt \rightarrow x_{k+1} - x_k \approx \frac{1}{2}(h_k)(f(x_{k+1}, u_{k+1}) + f(x_k, u_k)))$. This is called the trapezoid rule for integration where $h_k = t_{k+1} - t_k$ and $k = 1, 2 ... N_{grid} - 1$. The trapezoidal rule is also used to discretize the cost function. The constraints and boundaries are imposed in the respective points.

The result of the transcription in a trapezoidal direct collocation method is a nonlinear optimization problem with: a discretized cost function (trapezoidal rule), $n_x \times (N_{grid} - 1) + n_u \times (N_{grid} - 1)$ decision variables (corresponding to the discretization of the trajectory, n_x size of the state vector and n_u size of the input vector); each point subject to collocation, state and input constrains; and boundary constraints.

5.1. Boundaries and Constraints - Moonhopper

For the Moonhopper system it is required to define some state constraints that represent part of its dynamics. Regarding the time of jumping it was seen that it is equal to $5.3s$. In this case, a $5s$ final boundary is imposed. Regarding state variables, the values for ψ, θ, $\dot{\psi}$ and $\dot{\theta}$ are constrained. According to the data sheet, $-\pi/2 < \psi < \pi/2$, $-\pi/2 < \theta < \pi/2$, $-10 < \dot{\psi} < 10$, $-8.7 < \dot{\theta} < 8.7$. The gimbal inputs must lie inside the range $-0.2 < u_{\psi,\theta} < 0.2$.

Initial and final boundaries are determined by the initial and desired states, respectively. These are defined for each maneuver individually.

Finally, the cost function used corresponds to $\int_{t_0}^{t_F} \left(\ddot{\psi}^2 + \ddot{\theta}^2 + \ddot{\Omega}^2 \right) d\tau$. This was chosen as it reached the results faster and enforced the solution to have a smooth behaviour leading to less oscillations while penalizing high energy cost trajectories.

6. Results & discussion

In this section, the results of the experiments performed are presented. The first result corresponds to the model validation. For validating the model, a simulation of the model is run in a 3D robotics simulator framework ROS/Gazebo. Then, results are compared with the simulation of the state space dynamics described in section 2 and run in Matlab's ode45 solver. The second part of the results comprise a comparison of two control systems described earlier: the classic attitude controller and the open loop trajectory optimization based controller. For comparing these trajectories, an observation maneuver was designed. Upon take off the robot departs with an attitude of $[0, 0, 0]$ (roll, pitch and yaw) and an arbitrary initial angular velocity caused by disturbance torques. Before reaching the highest point of its trajectory, the robot must be able to achieve an orientation that enables an observation over the target area. Using an attitude hold controller, the actuators hold this orientation during one second (enough time to perform measurements and collect images), while the robot reaches the highest point and starts the downward trajectory. During this final phase, the system must be able to achieve a safety attitude described by safety conditions.

6.1. Model Verification

In general, a ROS/Gazebo model is defined by a set of links and joints commanded by transmissions. In addition, a series of plugins can be included to perform measurements as sensors, or vision capabilities as cameras. A model of the robot Moonhopper was defined using information regarding the inertia of the prototype and motion of the different parts of the robot structure (specified as joints). Several tests were performed, each with the goal to represent a characteristic of the system and highlight actuation properties. However, due to limited space, only a test that involve the three inputs u_G, u_H and u_W is performed. The maneuver starts with an acceleration of the wheel caused by a step in u_W. When $t = 5s$ the input signals for u_G and u_H are started. Figures (3-7) present the results for the designed test. The signals presented in the color red correspond to the state space model simulated in Matlab. The color blue is used to represent the results from the ROS/Gazebo model.

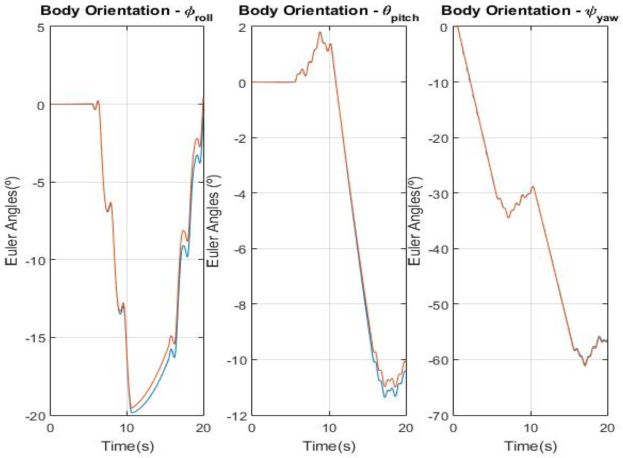

Figure 3: Outer gimbal-Inner gimbal step - Orientation in Euler Angles. The red line represents values of the orientation output from the state space model. The blue line represents values from simulation. The same applies for Figures (3-6).

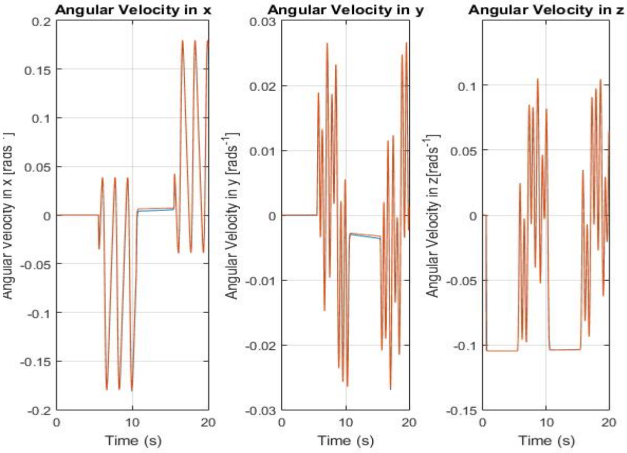

Figure 4: Outer gimbal-Inner gimbal step - Orientation in Euler Angles.

6.1.1 Discussion

As a result of these experiments it is possible to observe that, in general, the outputs of the model match the simulation outputs. In fact, it is seen that the errors are quite small for most of the cases and intuitively, both systems (Gazebo and State Space) behave like what is expected.

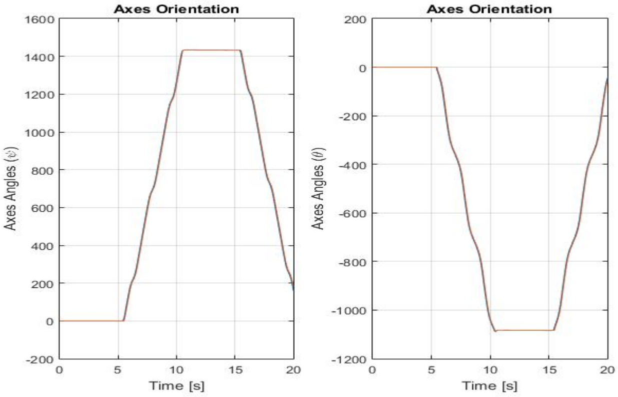

Figure 5: Outer gimbal-Inner gimbal step - Orientation in Euler Angles.

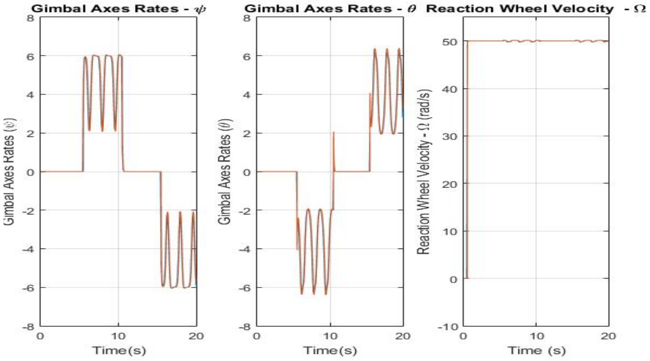

Figure 6: Outer gimbal-Inner gimbal step - Joint angular velocity

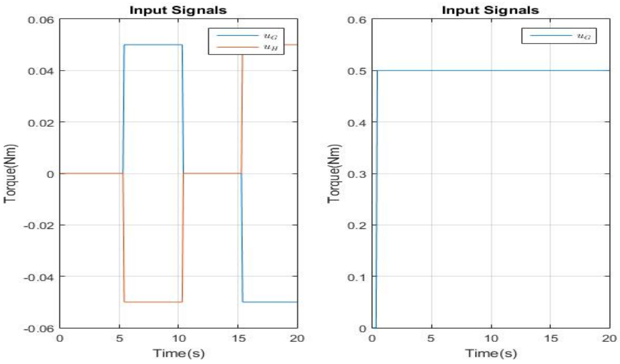

Figure 7: Input torque signals - Gimbal torque inputs u_G and u_H in Nm (left) and wheel torque input u_W in Nm (right)

It is seen that for the ψ_{roll} and θ_{pitch} angle signals (Figure 3), the error between the two models increases after $t = 15s$. However, the shape of the signal remains similar although offset by a fixed error. In fact, it was seen that these errors correspond to the gazebo model. Since this robot is relatively small when compared to the robots usually simulated in Gazebo, this framework results present errors due to simplifications and truncation. For larger robots (larger inertia) these errors are not as common. Additionally, it is important to mention that both models were forced to have friction introduced to the joints as the framework was not expected to perform well with ideal joints (no friction). By increasing the friction coefficient in joints, the error between both models decreased.

In conclusion, the results prove the validation of the state model described in section 2.

6.2. Moonhopper Attitude Control

The maneuver here presented starts with initial attitude of $[0, 0, 0, 1]$ (q_x, q_y, q_z and q_w). When $t = 2s$, the robot must reach an orientation of $\phi_{roll_r} = -55$º, $\theta_{pitch_r} = 40$º and $\psi_{yaw_r} = 40$º and remain with this orientation for one second. Finally, when $t = 3s$ it starts the attitude regain, achieving a safe attitude and angular velocity for landing, before $t = 5s$. This safe landing state is defined by an attitude between -3º and 3º in the angles of roll and pitch (aligned with $\hat{n}_1$ and $\hat{n}_2$ according to Figure 2) and an angular velocity ω between $-0.02\, rad\, s^{-1}$ and $0.02\, rad\, s^{-1}$.

6.2.1 Classical attitude control

The first controller to be implemented is the classical attitude approach. A trajectory connecting the attitude goals is first designed. Then, the controller is used to obtain the inputs to follow the trajectory. The result is presented in terms of the states (ω, ψ, θ, Ω, $\dot{\psi}$, $\dot{\theta}$) and the inputs in the form of $\ddot{\psi}$, $\ddot{\theta}$ and $\dot{\Omega}$ and in the form of u_G, u_H and u_W. Although the dynamics are used in q, the Euler angles representation is here presented for a more intuitive understanding of the results. Figures 8 to 11 present the trajectory

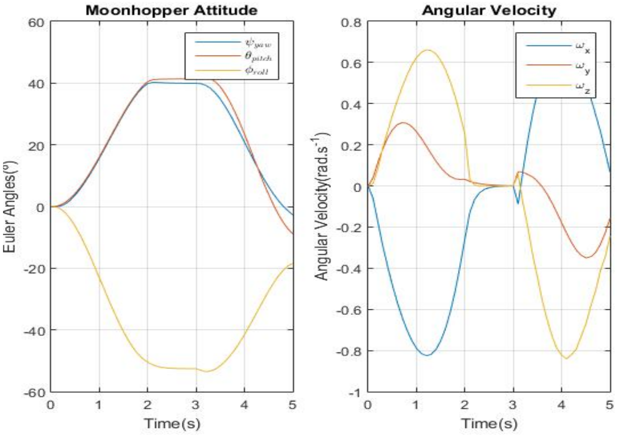

Figure 8: Observation test: Classic attitude controller - Attitude and

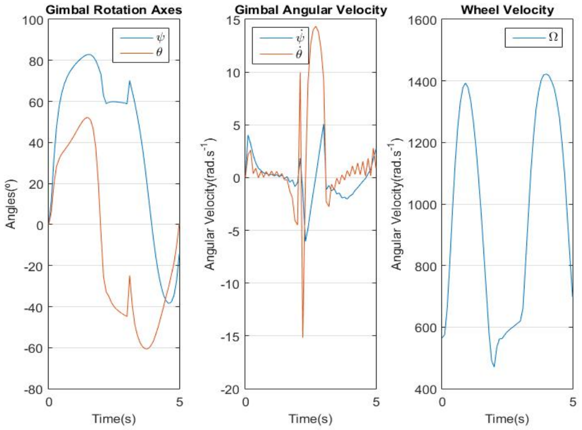

Figure 9: Observation test: Classic attitude controller - Gimbal Axes and Wheel velocity

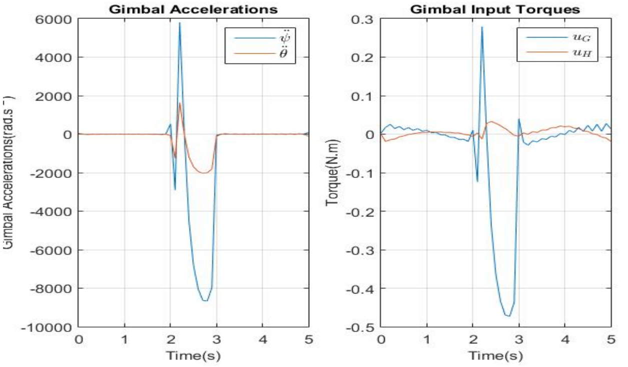

Figure 10: Observation test: Classic attitude controller - Gimbal inputs

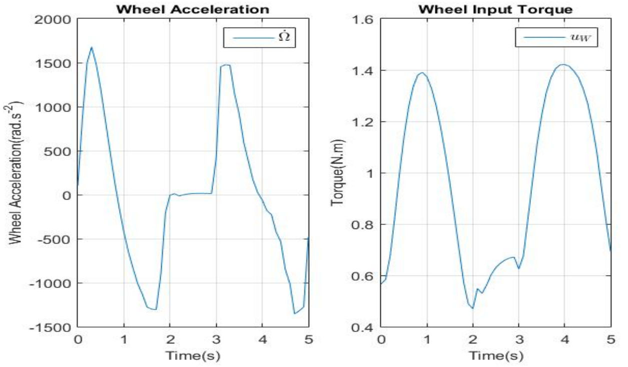

Figure 11: Observation test: Classic attitude controller - Wheel inputs

6.2.2 Trajectory optimization based control

In order to apply trajectory optimization to the observation maneuver, the $5s$ observation trajectory is partitioned in three sub trajectories defined by an initial state and an objective state. The first sub trajectory lasts $2s$, starts with the robots initial trajectory and finishes when the robot reaches the desired attitude for observation. The second sub trajectory corresponds to a 1 second attitude hold and is carried out by an attitude holder controller. When $t = 3s$, the last sub trajectory follows the same as for the classic controller. Only the first and the last sub trajectories are subjected to trajectory optimization, as the second is carried out by a developed controller.

The first objective sub trajectory has the initial state boundary $q_0 = [0, 0, 0, 1]$ and $\omega_0 = [0, 0, 0]$ and the final state boundary $q_F = [-0.340, 0.439, 0.118, 0.823]$ (corresponding to the attitude objective) and $\omega_F = [0, 0, 0]$. The last sub trajectory has the initial state boundary corresponding to the state at $t = 3s$ and the final boundary state $q_F = [0, 0, 0, 1]$. The remaining boundaries are presented in section 5. The trajectory optimization problem was introduced in the solver OptimTraj [4] and the outputs are presented in Figures 12 to 17.

6.2.3 Discussion

The first observation to be mentioned is the accomplishment of all objectives both for the attitude control and trajectory optimization controller. In fact, the de-

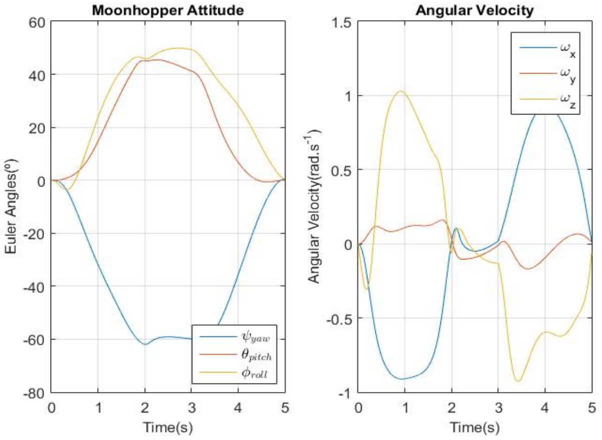

Figure 12: Observation test: Trajectory optimization controller - Attitude and angular velocity

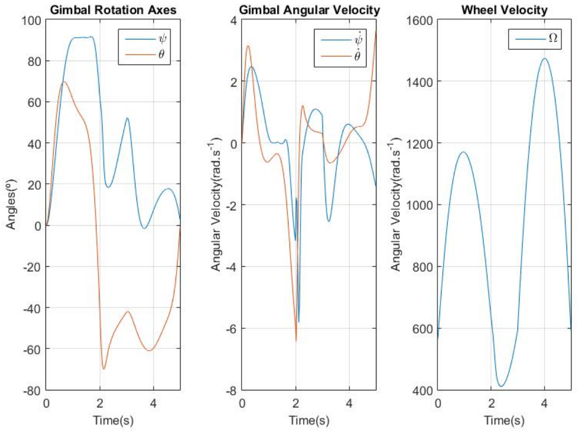

Figure 13: Observation test:Trajectory optimization controller - Gimbal Axes and Wheel velocity

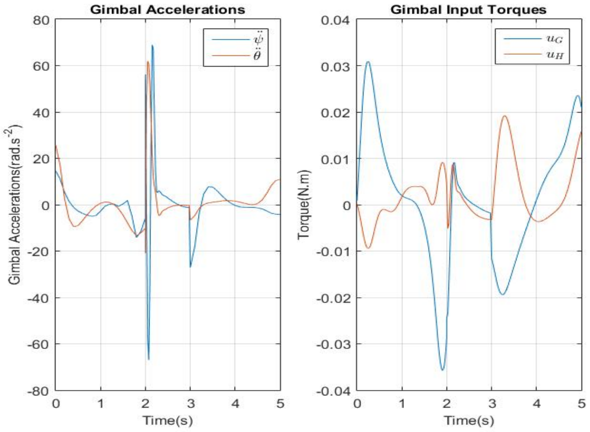

Figure 14: Observation test: Trajectory optimization - Gimbal inputs

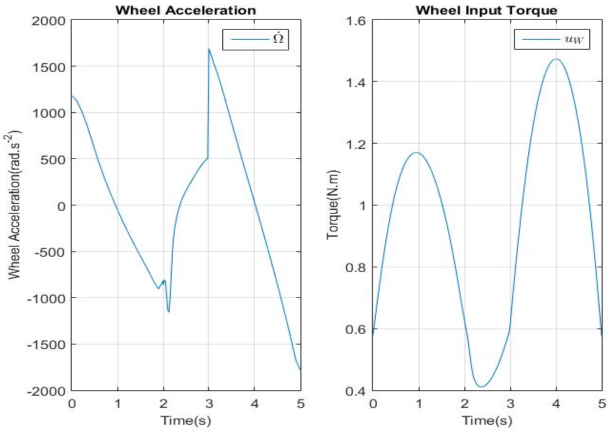

Figure 15: Observation test: Trajectory optimization - Wheel inputs

sired attitude is achieved and hold with a small angular velocity, almost zero. These conditions are ideal for capturing images. Regarding the gimbal axes it is clear the impact they have on the attitude control. However, this is done at a high cost in terms of velocity of the gimbal axes which are translated in high amplitude signals over a short period of time. This is more clear for the transition that occurs at $t = 2s$. Nevertheless, in general, the gimbal angular velocity remains inside the available range. The other notable observation is the extremely high gimbal accelerations, specially during the attitude hold phase. The reason for that is related to the small inertia matrices of the gimbals. Inverting these inertia matrices leads to high acceleration inputs which causes this type of behaviour. Although gimbal accelerations values are quite high, the input torque are actually small in comparison, reaching values between -0.4 and 0.2 N.m for the worse case scenario. For the trajectory optimization maneuvers, the wheel speed is considerably smaller than the same for the classic approach controller. In addition, the curves of $\dot\Omega$ are considerably different. This results in a relaxed maneuver, where only the focus is to achieve the desired state. This results in less rigid control over the ψ_{yaw} angle. The result is a lower energy consumption associated with this maneuver. Results show that for accomplishing this maneuver, the energy consumption is about 10% less for the trajectory optimization based controller when compared to the classical approach. For simpler maneuvers, like locomotion, results show less consumption. Finally, it should be noted the high velocity of the wheel. In fact, the reason lies on the small inertia of this component. Since the trajectory starts with an initial velocity of the wheel, the amount of initial angular momentum is considerably small for the robot $mW = 0.08857\,kg$ and $I_{wzz} = 1.0334e-5$.

6.2.4 Nonlinear control approach

The trajectories obtained from the trajectory optimization are used as input for a nonlinear controller [5] applied to the Moonhopper dynamics. These inputs are presented in the form of attitude trajectories in terms of quaternions and angular velocities.

Comparing these results with the open loop system it is seen that they are both similar in all states. In addition, the result for the energy consumed during the maneuver is similar to the open loop controller. This result leads to conclude that a combination of the two concepts is possible. The robustness associated with a closed loop controller gathered with the efficiency of the pre planned trajectories present a appropriate

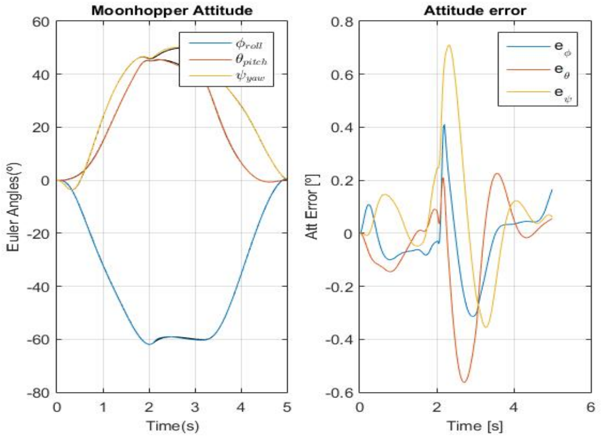

Figure 16: Observation test: Nonlinear control+optim trajectories

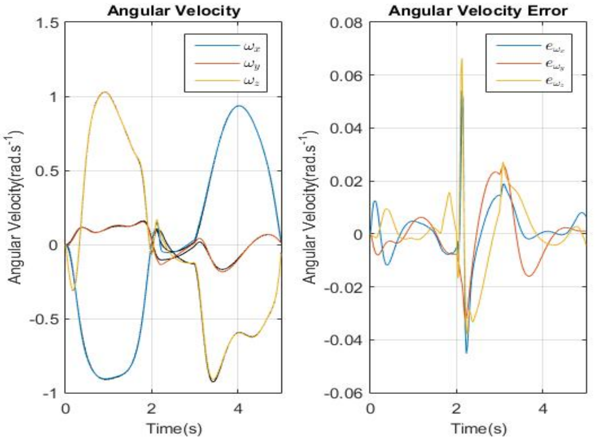

Figure 17: Observation test: Nonlinear control + optim trajectories

solution for the attitude control problem of the robot.

7. Conclusions

As it was stated in the first chapter of this document, the objective of this thesis is to study the solution for an attitude controller based on the use of a DGVSCMG. This was accomplishes by an initial characterization of the model dynamics regarding the translation and rotation motions. The resulting state space system was used for developing the three controllers used in this approach: the (PD+dynamic inversion), the trajectory optimization based open loop and the nonlinear Lyapunov controllers. By comparing the first two controllers it was possible to conclude the advantages regarding energy consumption of the trajectory based controller. For that reason, the trajectories were used to be followed by the nonlinear controller. The result lead to the description of the final control, that combines robustness and efficiency. It is important to note that the applications of this results are greater than this project. In fact, the both the model and the controllers are valid for an body actuated by a DGVSCMG.

References

[1] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[2] S. P. Bhat and P. K. Tiwari. Controllability of spacecraft attitude using control moment gyroscopes. *IEEE Transactions on Automatic Control*, 54(3):585–590, 2009.

[3] J. Burdick and P. Fiorini. Minimalist jumping robots for celestial exploration. *International Journal of Robotics Research*, 22:653–674, 07 2003.

[4] M. Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

[5] D. Stevenson and H. Schaub. Nonlinear control analysis of a double-gimbal variable-speed control moment gyroscope. *Journal of Guidance, Control, and Dynamics*, 35(3):787–793, 2012.

[6] R. Tammepõld, P. Fiorini, and M. Kruusmaa. Attitude control of small hopping robots for planetary exploration: theory and simulations. *ESA/ESTEC, Netherlands*, pages 1–8, 2011.

[7] Z. Zhang, J. Zhao, H. Chen, and D. Chen. A survey of bioinspired jumping robot: Takeoff, air posture adjustment, and landing buffer. *Applied Bionics and Biomechanics*, 2017:1–22, 09 2017.

8. Appenix

The details of the implementation are presented in this section regarding the details of the robot structure and the details of the controllers.

Inertia of the components:

Leggs - $m_L = 0.261$, $r_L = [0, 0, 0.05]^T$

$I_{L_xx} = 0.0013$; $I_{L_yy} = 9.8334\times10^{-4}$; $I_{L_xy} = 2.9174\times10^{-6}$; $I_{L_xz} = -5.7147 \times 10^{-5}$; $I_{L_zz} = 9.0571 \times 10^{-4}$; $I_{L_yz} = -2.9073 \times 10^{-6}$;

Body - $m_B = 0.82$ $r_B = [-0.002, 0.005, 0.25]^T$

$I_{B_xx} = 0.0017$; $I_{B_yy} = 0.0165$; $I_{B_xy} = 9.7637\times10^{-05}$; $I_{B_xz} = 7.0803 \times 10^{-05}$; $I_{B_zz} = 0.0028$; $I_{B_yz} = -2.9263 \times 10^{-05}$;

CMG Holder - $m_S = 0.20982$ $r_S = [0, 0, 0.28]^T$

$I_{S_xx} = 4.4590 \times 10^{-4}$; $I_{S_yy} = 4.4590 \times 10^{-4}$ $I_{S_zz} = 0.0011$;

Outer Gimbal (G) - $m_G = 0.02207$ $r_G = [0, 0, 0.345]^T$

$I_{G_xx} = 4.2974\times10^{-5}$ $I_{G_yy} 4.3426\times10^{-5}$ $I_{G_zz} 8.5739\times 10^{-5}$

Inner Gimbal (H) - $m_H = 0.020$ $r_H = [0, 0, 0.345]^T$

$I_{H_xx} = 1.1264 \times 10^{-5}$ $I_{H_yy} = 1.4680 \times 10^{-6}$ $I_{H_zz} = 1.1713 \times 10^{-5}$

Wheel (W) - $m_W = 0.00857$ $r_W = [0, 0, 0.345]^T$

$I_{W_xx} = 5.1682 \times 10^{-6}$ $I_{W_yy} 5.1682 \times 10^{-6}$ $I_{W_zz} = 1.0334 \times 10^{-5}$

Regarding the controller, the gain that were used are $K = 0.5$, $P = diag([0.1\ \ 0.1\ \ 0.1])$ $K_{\dot\psi} = 0.2$ $K_{\dot\theta} = 0.2$