Analysis, Implementation, and Simulation of the Cubli 3D Inverted Pendulum

Alessandro Mirante (1611555)

Cecilia Aponte Peredo (1822225)

Cesar Mazquiaran (1795330)

September 17, 2019

ABSTRACT

This paper addresses the analysis and implementation of the control law proposed in the paper "Nonlinear Analysis and Control of a Reaction Wheel-based 3D Inverted Pendulum" which allows the Cubli to maintain balance on a corner. The implementation and simulation animation are setup using MATLAB.

1 INTRODUCTION

In order to achieve efficient motions, plan and control methods are analyzed using a system's dynamics. One of those systems are those underactuated, which has less actuators than degrees of freedom. An example of such underactuated system is the Cubli which is a 3D inverted pendulum. It contains 3 reaction wheels (actuators) and higher-degrees of freedom. These reaction wheels rotate at high angular velocities and then suddenly stop in order make the Cubli jump up on its own from a resting position. It then remains balanced in that unstable, upright equilibria, as shown in Figure 1, regardless of external disturbances. To make this possible, the controller is designed with a mechanical perspective which uses backstepping for balancing and feedforward and linear state feedback for the jump up. Resulting in a smooth, asymptotically stable control law.

The dynamics of the system using Lagrangian and Lyapunov are analyzed and described in Section II, followed by the non-linear controller and experimental results from the original paper in Section III, and lastly our implementation in MATLAB, its animation, and experimental results in Section IV.

Figure 1: Cubli balancing on its corner

2 SYSTEM DYNAMICS AND ANALYSIS

REACTION WHEEL INVERTED PENDULUM 1D

To start the modeling of the Cubli, we have to talk about its nature, as it is a cube of 150 mm side length with three reaction wheels mounted orthogonally to each other. So first we have to get some insight about a reaction wheel based inverted pendulum, which is the component that will lead the system to its equilibrium point.

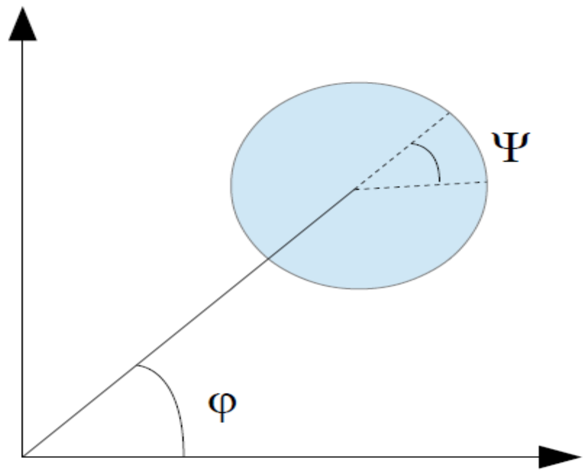


Figure 1: Schematic diagram of a reaction wheel based 1D inverted pendulum.

Where the angles φ and ψ denote the position of the 1D inverted pendulum. Now, lets denote the reaction wheel's moment of inertia by Θ_w, and the system's total moment of inertia around the pivot point in the body fixed coordinate frame by Θ_o, the total mass as m_{tot} and finally l is going to be used for the distance from the pivot point to the center of gravity of the whole system.

The Lagrangian of the system is going to be:

$$\begin{gathered}\mathrm{L{=}K{+}V{=}}\tfrac{1}{2}\hat{\Theta}_o\dot{\varphi}^2+\tfrac{1}{2}\hat{\Theta}_w(\dot{\Psi}+\dot{\varphi})^2-\mathrm{mgcos}\,(\varphi)\\ \hat{\Theta}_o=\Theta_o-\Theta_w\\ \mathrm{m{=}m_{tot}}l\end{gathered} \tag{1}$$

To get this Lagrangian approach for the dynamics, we have to take into account the kinematic and potential energies of the system. The kinematic energies of the system are shown in the first two terms, and the last term is the potential energy.

If we now take the derivatives of the Lagrangian to obtain the generalize momenta, we obtain:

$$p_\varphi := \frac{\partial L}{\partial \dot{\varphi}} = \hat{\Theta}_o\dot{\varphi} + \Theta_w(\dot{\Psi}+\dot{\varphi}) = (\Theta_o-\Theta_w)\dot{\varphi} + \Theta_w(\dot{\Psi}+\dot{\varphi}) = \Theta_o\dot{\varphi} + \theta_w\dot{\varphi} \tag{2}$$

Now, we have to introduce a new term, T , which represents the applied torque to the reaction wheel, so we can derive the equations of motion using the Euler-Lagrange approach, with T as a non potential force.

$$\begin{gathered}\dot{p}_\varphi := \frac{\partial L}{\partial \varphi}\mathrm{{=}mgsin}\,(\varphi)\\ \dot{p}_\Psi := \frac{\partial L}{\partial \Psi}\mathrm{{+}T{=}T}\end{gathered} \tag{3}$$

Thanks to the introduction of the generalized momenta, equations in (2), we have a more simple representation of the system, as we can see in (3), where we have the usual inverted pendulum augmented by an integrator.

Last but not least, we have to recall that the position of the reaction wheel is not important, so we can write the reduced set of states of our dynamics on the following form:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{p}_{\varphi} \\ \dot{p}_{\Psi} \end{pmatrix} = \mathrm{f}(x, T) = \begin{pmatrix} \hat{\Theta}_o^{-1}(p_{\varphi} - p_{\Psi}) \\ mgsin(\varphi) \\ T \end{pmatrix} \tag{4}$$

Where the first term can be obtained isolating and substituting one term into the other in (2).

Analysis

For the following analysis, the state space is such that X={x | φ ϵ (-π , π], p_{φ} ϵ $\mathbb{R}$, p_{Ψ} ϵ $\mathbb{R}$ }.

To obtain the equilibrium points of our system, we have to look for the points such that : ε={(x,T) $\epsilon X \times \mathbb{R}$ | f(x,T) = 0}. Evaluating the dynamics equal to 0, we obtain two equilibrium points:

$$\begin{aligned} \varepsilon_1 &= \left\{ (\mathrm{x,T})\, \epsilon \mathrm{X} \times \mathbb{R} | \varphi{=}0, \mathrm{p}_{\varphi}{=}\mathrm{p}_{\Psi}, \mathrm{T}{=}0 \right\} \\ \varepsilon_2 &= \left\{ (\mathrm{x,T})\, \epsilon \mathrm{X} \times \mathbb{R} | \varphi{=}\pi, \mathrm{p}_{\varphi}{=}\mathrm{p}_{\Psi}, \mathrm{T}{=}0 \right\} \end{aligned} \tag{5}$$

ε_1 represents the upright unstable equilibrium point as the angle φ=0, and ε_2 represents the hanging equilibrium point as $\varphi = \pi$. The first equilibrium point means that the system can be at rest, while we maintain a constant angular velocity in the reaction wheel.

REACTION WHEEL INVERTED PENDULUM 3D

As we said before, the Cubli is composed of three reaction wheel inverted pendulums mounted orthogonally between them.

In the figure number 2, B_e and I_e denote the principle axis of the body fixed frame {B} and inertial frame {I} The pivot point O is the common origin of coordinate frames {B}and {I} .

As in the case of a single reaction wheel inverted pendulum, now we have to define the system's total moment of inertia around the pivot point in the body fixed coordinate frame by Θ_o, but we have to take into account that now the reaction wheel's moment of inertia will be a diagonal matrix with each of the reaction wheel's moment of inertia in the main diagonal: Θ_w =diag(Θ_{w1},Θ_{w2},Θ_{w3}). Last but not least, we will indicate the position vector from the pivot point to the center of gravity multiplied by the mass with $\vec{m}$ and the gravity vector with $\vec{g}$.

Before going deep on the analysis, first we have to clarify some concepts about notation. The projection of a tensor onto a particular coordinate frame is denoted by a preceding superscript, such us ${}^{B}\vec{m} = {}^{B}m \in \mathbb{R}^{3x3}$. As the body fixed coordinate frame, {B}, is the most commonly projected coordinate frame, we will usually remove the preceding superscript to make it more simple.

Starting now with the analysis of the dynamics, the Lagrangian of the system is now given by:

$$L\left(w_h, \mathrm{g}, \mathrm{w}_w, \Phi\right) = \frac{1}{2} w_h^T \hat{\Theta}_o w_h + \frac{1}{2}\left(w_h + \mathrm{w}_w\right)^T \Theta_w \left(w_h + \mathrm{w}_w\right) + \mathrm{m}^T g \tag{6}$$

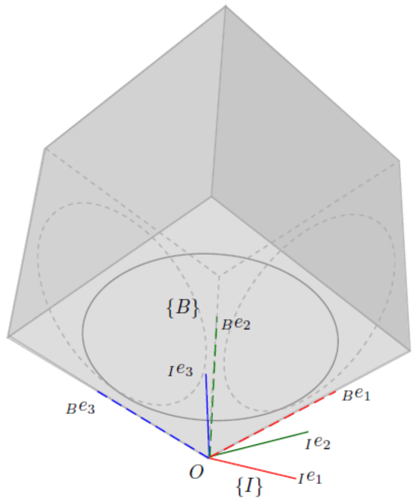


Figure 2: Cubli balancing on the corner

This Lagrangian can be interpreted again as the single inverted pendulum as the sum of all kinetic and potential energies. To obtain the first set of these, the Konig Theorem can be applied. We also have to take into account that the idea of removing the superscripts has been done in both the body angular velocity and reaction wheel angular velocity.

The components of the torque applied to the reaction wheels are inside $T \ \epsilon \ \mathbb{R}^3$. The components of the vector *m* are constant in the body fixed frame {B}, while the gravity is not:

$$^{B}\dot{\vec{g}} = \dot{g} + \mathrm{w}_h \times g = \ \dot{g} + \tilde{w}_h g = 0 \tag{7}$$

This tilde operator is such that applied to a vector $v \ \ \epsilon \ \mathbb{R}^3$, it denotes the skew-symmetric matrix for which the following is true: $\tilde{v}$a=v $\times \ a$ holds $\forall a \ \mathbb{R}^3$.

Following the same steps than in the simple reaction wheel inverted pendulum in a plane, we an derive now the generalized momenta as:

$$\begin{aligned} p_{w_h} &:= \frac{\partial L}{\partial w_h}^T = \Theta_o w_h + \Theta_w w_w \\ p_{w_w} &:= \frac{\partial L}{\partial \dot{\Psi}}^T = \Theta_w \left(w_h + \mathrm{w}_w \right) \end{aligned} \tag{8}$$

From here, we can derive the expressions of the body angular velocity $\mathrm{w_h}$: $w_h = \hat{\Theta}_o^{-1} \left(p_{w_h} - p_{w_w} \right)$, and the equations of motion too:

$$\begin{aligned} \dot{g} &= -\tilde{w}_h g \\ \dot{p}_{w_h} &= -\tilde{w}_h p_{w_h} + \tilde{m} g \\ \dot{p}_{w_w} &= \mathrm{T} \end{aligned} \tag{9}$$

The last step, if we use $\dot{\vec{v}} = \dot{v} + \mathrm{w}_h \times v$, the time derivative of a vector in a rotating frame, equations in (9) can be further simplified to:

$$\begin{aligned} \dot{g} &= 0 \\ \dot{\vec{p}}_{w_h} &= \vec{m} \times \vec{g} \\ \dot{p}_{w_w} &= \mathrm{T} \end{aligned} \tag{10}$$

Which is useful to highlight the similarity between the 2D and 3D inverted pendula: $\|\dot{\vec{p}}_{w_h}\|_2 = \|\vec{m}\|_2\|\vec{g}\|_2 \sin(\Phi)$ Where Φ is the angle between $\vec{m}$ and $\vec{g}$. Moreover, p_{w_w} is the integral of the applied torque *T*.

Analysis

First thing to contemplate is the conservation of the angular momentum, and as we can see in (10), this is done as the rate of change of the momentum of the body is orthogonal to $\vec{m}$ and $\vec{g}$ at any moment. Since the latter is constant, $\vec{p}_{w_h}$ will never change its component in the direction of the gravity vector. If we talk in the body fixed frame {B}, this concept of conservation of the angular momentum around $\vec{g}$ can be written as:

$$\frac{d}{\mathrm{dt}}\left(p_{w_h^T} g\right) \equiv 0 \tag{11}$$

The importance of this is the following: Independently of the control input chosen, the monument around $\vec{g}$ is conserved, and it may be unachievable to reach an equilibrium depending on the initial conditions.

The state space of the 3D Cubli will be similar to the simple inverted reaction wheel inverted pendulum in 2D. Taking into account that the analysis will be performed in the fixed body coordinate frame, the state space will be the set: $\mathrm{X}=\left\{\mathrm{x}=\left(\mathrm{g},\mathrm{p}_{w_h},\mathrm{p}_{w_w} \in \mathbb{R}^9 \mid \|g\|_2 = -9{,}81\right)\right\}$

The next step is to talk about the equilibrium points, and again it follows a similar procedure. Setting the right-hand equations of (8) and (9) to zero it leads to:

$$\begin{gathered} -\bar{w}_h \times \bar{p}_{w_h} + \mathrm{m} \times \bar{g} = 0 \\ -\bar{w}_h \times \bar{g} = 0 \\ \bar{T} = 0 \end{gathered} \tag{12}$$

Where $\bar{w}_h, \bar{p}_{w_h}, \bar{g}$ are the equilibrium configurations of the system.

If we analyze the second equation of (12) it shows us that $\bar{w}_h || \bar{g}$, or in another words: $\bar{w}_h = \lambda_1 \bar{g}$, $\lambda_1 \in \mathbb{R}$. Thus, the relative equilibria are denoted by:

$$\begin{gathered} \bar{w}_h = \lambda_1 \bar{g} \\ \lambda_1 \bar{p}_{w_h} + \mathrm{m} = \lambda_2 \bar{g} \\ \bar{T} = 0 \end{gathered} \tag{13}$$

This denotes again two equilibria, hanging and upright positions. The hanging one is stable and the upright is unstable. This is obtained setting λ_1=0, which implies $\bar{w}_h || \bar{g}$.

3 NONLINEAR CONTROLLER AND ORIGINAL RESULTS

There are two different control strategies to asymptotically stabilize the upright equilibrium. The first is based on backstepping which smooth, globally stabilize used for balancing. The second is based on feedback linearization used for tracking predefined non-equilibrium motions.

To build the control law, some simplifications are established. Since the state space is chosen to be

$$(g, p_{w_h}, p_{w_w}) \ \in := S^2 \times \mathbb{R}^3 \times \mathbb{R}^3, \tag{14}$$

the feedback control laws will not change based on the orientation around gravity and the reaction wheel positions. The component of the angular momentum p_{w_h} in the direction of gravity is conserved, therefore only its component orthogonal to g can be affected by feedback control. p_{w_h} is then split into two parts: one in the direction of gravity $p^g_{w_h}$ and one orthogonal to it $p^{\perp}_{w_h}$. Resulting in the following

$$\begin{aligned}
\vec{p}_{w_h} &= \vec{p}^{\perp}_{w_h} + \vec{p}^{g}_{w_h} \\
\vec{p}^{g}_{w_h} &= (\vec{p}^{T}_{w_h})\frac{\vec{g}}{||\vec{g}||^2_2} \\
\dot{\vec{p}}^{\perp}_{w_h} &= \dot{p}^{\perp}_{w_h} + \tilde{w}_h p^{\perp}_{w_h} = \tilde{w}_h mg
\end{aligned} \tag{15}$$

The control objective for the Cubli is to balance in the upright position and that w_h and $p^{\perp}_{w_h}$ would approach zero as time goes to infinity. In order for this control objective to prove asymptotic stability, it must exclude the hanging equilibrium

$$x^{-} = x \in X \ | \ g = \frac{||g||_2}{||m||_2}, \ p^{\perp}_{w_h} = 0, \ p_{w_h} = p_{w_w} \tag{16}$$

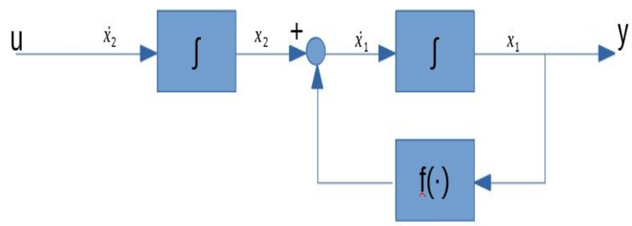
3.1 Backstepping Approach

For some systems, there exist an iterative procedure to get a Lyapunov function. Lets imagine we have a system on the form:

$$\begin{aligned}
\dot{x}_1 &= f(x_1) + x_2 \\
\dot{x}_2 &= u \\
y &= x1
\end{aligned} \tag{17}$$

The equilibria around the origin will be on the form:

$$Eq = \begin{bmatrix} 0 \\ -f(0) \end{bmatrix} \tag{18}$$



As we want to stabilize the output y $\rightarrow 0$, first thing we should do is to "cut" the system after the first integrator, like we can see on the previous image.

1) Define:

$$\begin{aligned} z_1 &= x_1 \\ \dot{z}_1 &= f(z_1) + \alpha_1 \end{aligned} \tag{19}$$

Which is a different system to the given system.

Now we have to stabilize, so we have to get an $\alpha_1 = -f(z_1) - k_1 z_1$ with $k_1 > 0$, and place it in $\dot{z}_1$. After this, it is time to find the first Lyapunov candidate, which usually will be on the form:

$$V_1(z_1) = \frac{1}{2} z_1^2 \rightarrow \dot{V}_1(z_1) = z_1 \dot{z}_1 \tag{20}$$

We discover that α_1 is not available, so we rewrite the first equation:

$$\begin{aligned} \dot{z}_1 &= f(z_1) + x2 + \alpha_1(z_1) - \alpha_1(z_1) \\ z_2 &= x_2 - \alpha_1 \end{aligned} \tag{21}$$

Where z_2 is an intermediate variable.

$$\begin{aligned} \dot{z}_1 &= f(z_1) + \alpha_1(z_1) + z_2 \\ \dot{z}_2 &= \dot{x}_2 - \dot{\alpha}_1 = u - \dot{\alpha}_1 \end{aligned} \tag{22}$$

2) Define the second Lyapunov candidate V_2:

$$V_2(z_1, z_2) = V_1(z_1) + \frac{1}{2} z_2^2 \tag{23}$$

According to the Lyapunov Theorem, to have stability, the candidate must be positive definite and its derivative equal or less than zero.

$$\dot{V}_2 = z_1 \dot{z}_1 + z_2 \dot{z}_2 = z_1 f(z_1) + \alpha_1(z_1) z_1 + z_1 z_2 + z_2 \dot{x}_2 - z_2 \dot{\alpha}_1 \tag{24}$$

Using the previous $\alpha_1 = -f(z_1) - k_1 z_1$, we have to find the control input u such that $\dot{V}_2$ is negative, taking into account that:

$$\dot{V}_2 = z_1 f(z_1) + (-f(z_1) - k_1 z_1) z_1 + (z_1 + u - \dot{\alpha}_1) z_2 \tag{25}$$

$$\dot{\alpha}_1 = -\frac{\partial f(z_1)}{\partial t} - k_1 \dot{z}_1 = -\frac{\partial f(z_1)}{\partial z_1} \dot{z}_1 - k_1 \dot{z}_1 = -(\frac{\partial f(z_1)}{\partial z_1} + k_1)(f(z_1) + \alpha_1 + z_1) \tag{26}$$

The obtained non linear control input u that stabilizes the system is:

$$u = -z_1 - k_1 z_1 - k_1 z_2 + NL = -z_1 - k_1 z_2 + \dot{\alpha}_1 \tag{27}$$

And the resulting system after the change of coordinates is such that:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 + f(x_1) + k_1 x_1 \end{bmatrix} \tag{28}$$

- Example of backstepping procedure

Now we will see an example on how to perform a stabilizing backstepping procedure using the following system:

$$\begin{aligned} \dot{x}_1 &= x_1^2 - x_1^3 + x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{29}$$

First of all, we must look for the zero dynamics f(x_1,0) :

$$\dot{x}_1 = x_1^2 - x_1^3$$
$$\text{Using now the Lyapunov candidate } V_{zd} = \frac{1}{2}x_1^2 \rightarrow \dot{V}_{zd} = x_1\dot{x}_1 = x_1^3 - x_1^4 \tag{30}$$

Which tells us that the system is not Asymptotically Stable for all values of $x_1 \neq 0$. For this reason, we have to perform a change of coordinates $x_2 = \phi(x_1) = -x_1^2 - x_1$, and now our system becomes:

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 = -x_1 - x_1^3 \tag{31}$$

Now we have to choose a first Lyapunov candidate to stabilize our system $V_1 = \frac{1}{2}x_1^2$. With this choice we have:

$$\dot{V}_1 = x_1\dot{x}_1 = -x_1^2 - x_1^4 \leq 0 \; \forall x_1 \tag{32}$$

To complete the change of coordinates: $z_2 = x_2 - \phi(x_1) = x_2 + x_1^2 + x_1$, which renders a "new" system on the form:

$$\dot{x}_1 = x_1 - x_1^3 + z_2;$$
$$\dot{z}_2 = \dot{x}_2 + 2\dot{x}_1 + x_1\dot{x}_1 = u + (1 + 2x_1)(-x_1 - x_1^3 + z_2) \tag{33}$$

The next step is to propose a new Lyapunov candidate: $V_2 = V_1 + \frac{1}{2}z_2^2 = \frac{1}{2}x_1^2 + \frac{1}{2}z_2^2$. Taking the derivative to study the stability we get:

$$\dot{V}_2 = \dot{V}_1 + z_2\dot{z}_2 = x_1\dot{x}_1 + z_2\dot{z}_2$$
$$\dot{V}_2 = x_1^2 - x_1^4 + x_1z_2 + z_2[u + (1 + 2x_1)(-x_1 - x_1^3 + z_2)] \tag{34}$$

Here now we see that our control input appears, and as to achieve stability $\dot{V}_2$ has to be such that $\dot{V}_2 \leq 0$, we have to choose u in order to achieve that:

$$u = -(1 + 2x_1)(-x_1 - x_1^3 + z_2) - x_1 + z_2$$
$$\text{using this u, our } \dot{V}_2 \text{ becomes:} \dot{V}_2 = -x_1^2 - x_1^4 - z_2^2 = -x_1^2 - x_1^4 - (x_2 + x_1^2 + x_1)^2 \; \forall x_1, x_2 \tag{35}$$

In our Cubli system, the backstepping approach is chosen to find a stabilizing controller. The result gives the following control law, where K_1 refers to the derivative gain, K_2 and K_3 are the proportional gains, and K_4 is the integral gain.

$$u = K_1\tilde{m}g + K_2w_h + K_3p_{w_h} - K_4p_{w_w} \tag{36}$$

where,

$$\begin{aligned}
K_1 &= (1 + \beta\gamma + \delta)I + \alpha\hat{\Theta}_0 \\
K_2 &= \alpha\hat{\Theta}_0\tilde{p_\omega}_h^\perp + \beta\tilde{m}\tilde{g} + \tilde{p}_{\omega_h} \\
K_3 &= \gamma(I + \alpha\hat{\Theta_0}(I - \frac{gg^T}{||g||_2^2})) \\
K_4 &= \gamma I \qquad \alpha, \beta, \gamma, \delta > 0
\end{aligned} \tag{37}$$

To understand the tunning parameters, they analyzed the closed-loop response in two different initial conditions. The first condition is when the Cubli is released at rest, but with a non-zero inclination angle. This was able to determine three of the tunning parameters: $\alpha, \beta,$ and δ. For the second condition, a pure yaw motion was analyzed and is related to the last tunning parameter γ.

This controller makes $x \in X \quad \chi$ asymptotically stable. The proof for this is that if one considers the Lyapunov candidate function V: $X \rightarrow \mathbb{R}$ found via a two-step backstepping approach.

$$V(x) = \frac{1}{2}\alpha p_{w_h}^{\perp}{}^T p_{w_h}^{\perp} + m^Tg + ||m||_2||g||_2 + \frac{1}{2\delta}z^T\Theta_0^{-1}z \tag{38}$$

This shows that there exists a K_∞ function a : [0, $\infty \rightarrow$ [0, ∞) such that $V(x = x_0) = 0$ and V(x) $\geq a(|x - x_0|)$ for all $x \in X$ and all $x_0 \in T$. This shows that V is a positive definite function and a valid Lyapunov function.

$\dot{V}$ is then evaluated along trajectories of the closed-loop system:

$$\begin{aligned} \dot{V}(x) &= \alpha \dot{p}^{\perp}_{w_h} + m^T \dot{g} + \frac{1}{\delta} z^T \hat{\Theta}_0^{-1} \dot{z} \\ &= -\beta(\tilde{g}m)^T \hat{\Theta}_0^{-1}(\tilde{g}m) - \frac{\gamma}{\delta} z^T \hat{\Theta}_0^{-1} z \end{aligned} \tag{39}$$

and since $\dot{V}(x) \leq 0$, $\quad \forall x \in X$ it can be concluded that the point $x_0 \in T$ is stable.

3.2 Feedback Linearization Approach

To control the nonlinear system, a feedback linearization is done to transform the nonlinear system into a linear one through a change of variables and control input. The system state space is given by (g, p_{w_h}, p_{w_w}), after the state transformation it is given by $x = (y, \dot{y}, \ddot{y}, {}^I w_{h_3})$. The linear system dynamics for the case ${}^I m_3 \neq 0$ as shown in [2] is given by

$$\dot{x} = \begin{pmatrix} 0_{2\times2} & I_{2\times2} & 0_{2\times2} & 0_{2\times1} \\ 0_{2\times2} & 0_{2\times2} & I_{2\times2} & 0_{2\times1} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & 0_{1\times2} & 0 \end{pmatrix} x + \begin{pmatrix} 0_{4\times3} \\ I_{3\times3} \end{pmatrix} \begin{pmatrix} w \\ {}^I v_3 \end{pmatrix} \tag{40}$$

To obtain this linearization, the generalized momentum p_{w_h} is chosen as the virtual output and it is projected to the intertial frame to remove the conserved component in the direction $\vec{g}$. The dynamics of the Cubli in the inertial frame is then as follows

$$\begin{aligned} {}^I \dot{m} &= {}^I w_h \times {}^I m \\ {}^I \dot{p}_{w_h} &= {}^I m \times {}^I g \\ {}^I \dot{p}_{w_w} &= {}^I T + {}^I w_h \times {}^I p_{w_w} \end{aligned} \tag{41}$$

Since the third component of ${}^I p_{w_h}$ is conserved, only the first two elements form the virtual output y as $y := ({}^I p_{w_h 1}, {}^I p_{w_h 2})$. To calculate the cross multiplication of the first two components, the following method is used

$$\begin{gathered} P(a \times b) = -a_3 JPb + b_3 JPa \\ \text{where,} \qquad J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{gathered} \tag{42}$$

Therefore, the calculation of the virtual output y is:

$$\begin{aligned} \dot{y} &= ({}^I \dot{p}_{w_h 1}, {}^I \dot{p}_{w_h 2}) = {}^I m \times {}^I g \\ &= P({}^I m \times {}^I g) = -|g| JP^I m \end{aligned} \tag{43}$$

Its corresponding derivatives are as follows

$$\begin{aligned} \ddot{y} &= -|g| JP^I ({}^I w_h \times {}^I m) \\ \dddot{y} &= -|g| JP^I ({}^I \dot{w}_h \times {}^I m + {}^I w_h \times {}^I \dot{m}) \end{aligned} \tag{44}$$

Also, the angular velocity of the body is given by

$$
\begin{aligned}
{}^{I}\dot{w}_h &= R_K \Theta_0^{-1}(\dot{p}_{w_h} - \dot{p}_{w_w}) \\
&= R_K \Theta_0^{-1}(m \times g - w_h \times p_{w_h} - T)
\end{aligned}
\tag{45}
$$

It follows that the equations are solved to the input torque T and using the change of variables to T to ${}^{I}v$ which leads ${}^{I}\dot{w}_h = {}^{I}v$. Also through some manipulation, it is found that $\ddot{y} = w$. This all obtains the final state transformation and linear system dynamics shown initially above.

3.3 Paper Results

To analyze this proposed controller and estimate the state, experiments were setup with disturbances of 0.1 Nm applied to each reaction wheel simultaneously. As seen below, the controller reaches steady state in < 0.3s for every component of the input $(g, \dot{p}_{w_h}, \dot{p}_{w_w})$ with some noise.

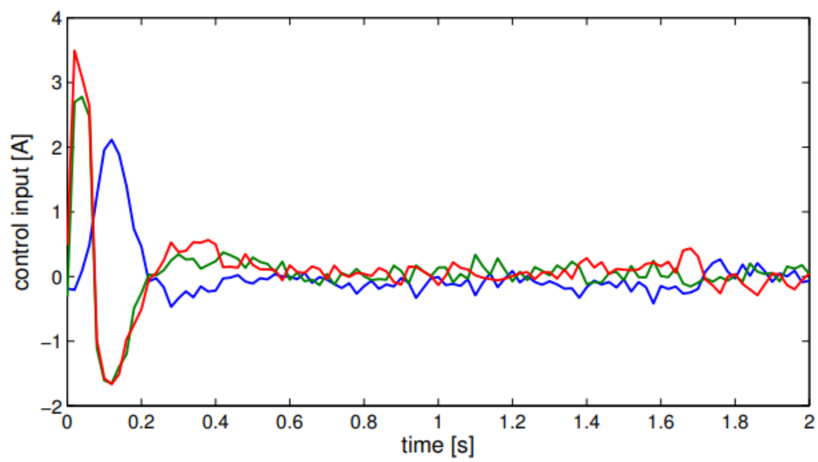


Figure 2: Results showing the time it takes to control the Cubli

The figure below shows the error in the inclination angle RMS with respect compared to that expected of the nonlinear controller is less than 0.025° at steady state.

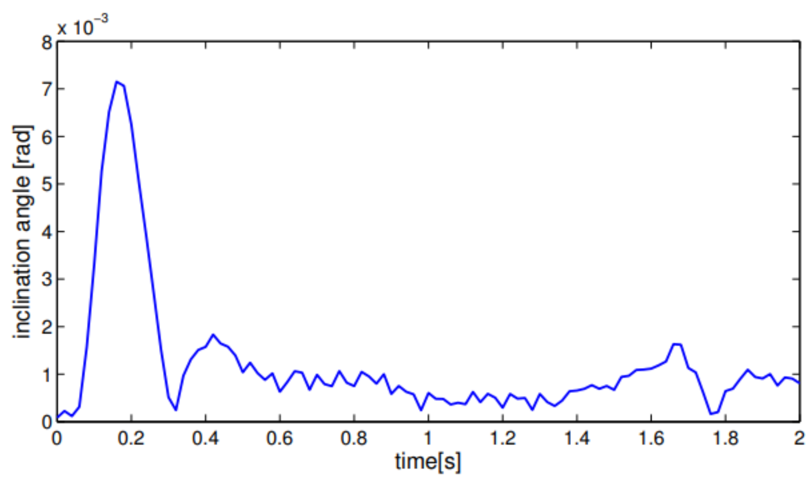


Figure 3: Results showing the error of the inclination angle of the Cubli

4 IMPLEMENTATION, ANIMATION AND RESULTS IN MATLAB

In this section is described the implementation and the discussion of a sequence of plots and animations, that will show different evolution starting from different Cubli's initial conditions. It's important to notice that Cubli controller uses as mass parameter, exactly the mass modelled in the dynamical system, negleting eventually unmodelled uncertainty. From the mechanical point of view the cube is considered to be hanged to one of its edge, the one on which the controller is trying to balance, in this way the contact force can be considered infinite. This control law was studied and implemented in Matlab, in addition we implement a 3D simulated animation of the behavior of the cube. This was accompanied by a sequence of different plots relating to the simulated Cubli's behaviour with various simulation conditions as initial conditions. The different equilibria objectives were examined to analyze the expected result of the upright equilibria as unstable and the hanging as stable in the sense of Lyapunov. The results obtained were as expected and will be described in the following subsections.

4.1 Introduction to Matlab Implementation

We choose to simulate the results concerning the control of a Simulated Cubli model on Matlab platform, using the integration of ordinary differential equations (ODE) that give us the evolution of the modeled system. For this kind of calculation it has been used a standard Solver for Matlab (ODE45) with variable steps of integration. The integration steps may change in function of the error generated during integration algorithm, in such a way that it's possible to get a reliable result.

```
%   Options ode

reltol = 1.0e-9;
abstol = 1.0e-9;

options = odeset('Reltol',reltol,'Abstol',abstol);

%
```

The Reltol is the tolerance related to this tolerance value and it's a measure of the error relative to the size of each solution component. Roughly, it controls the number of correct digits in all solution components, except those smaller than thresholds AbsTol.
AbsTol is a threshold below which the value of the i^{th} solution component is unimportant. The absolute error tolerances determine the accuracy when the solution approaches zero.

4.2 State Space Rapresentation

The state space rapresentation is divided in 13 elements, each element has it's own differential equation that we are going to describe. In order to avoid an overload of notation the vectors are written as scalar. The first three components describe the gravity vector, $\vec{g}$, which is an element strictly related to the attitude, in fact since g represents the gravity in the body frame, during the evolution is possible to see the direction of the vector moving is in function of the attitude, although we know that, from the inertial point of view, the vector gravity is assumed to be fixed. Indeed, with some mathematical manipulation we could extract complete attitute information, however we use another approach. The second group of three components is the angular momentum write as $\dot{p_{\omega_h}}$ in the pivot point on the vertex of the cube, while the last group is the angular momentum of the reaction wheels, $\dot{p_{\omega_w}}$,that is also the control input. Last remaining four components are the quaternions, used to describe the attitude w.r.t. the inertial frame. As we will describe after, with quaternions we have a simpler notation avoiding the use of rotation matrices, a procedure that involves also the use of the vector g, but maintaining its uniqueness of representation.

The model is described by those equations of motion:

$$
\begin{aligned}
&\dot{g} = -\tilde{\omega_h} g \\
&\dot{p_{\omega_h}} = -\tilde{\omega_h} g + \tilde{m} g \\
&\dot{p_{\omega_w}} = T \\
&\dot{q} = \frac{1}{2}\ \omega \times q
\end{aligned}
\tag{46}
$$

that in Matlab appears:

```
%------------------------------System Dynamic------------------------------

dpdt(1:3) =  -omega_h_skew*p(1:3);                           % gravity vector

dpdt(4:6) =  -omega_h_skew*p(4:6) + m_skew*p(1:3);           % momentum vertex

dpdt(7:9) = T;                                               % momentum wheels

q = [p(10),p(11),p(12),p(13)]';                              % actual Quaternions

%----------------------------------Skew Matrix 4x4-------------------------
Q = skew4x4(omega_h);
%--------------------------------------------------------------------------

dpdt(10:13) = 1/2 * Q * q;                                   % Quaternion's Attitude
```

We can notice that the vector product between ω and quaternions can be substituted with the product between skew symmetric matrix of ω_h and q, because we know that given two vectors a and b, the cross product $a \times b$ is equal to $\tilde{a}b$. Where $\tilde{a}$ is a skew-symmetric matrix.

4.3 Nonlinear Controller

The Nonlinear Controller developed using a backstepping procedure is a controller characterized by four tunning parameters. The value of those parameters are chosen once that a closed loop behavior has been related to as described in the reference [1].
The controller acts, weighting the state of Cubli following this law:

$$T = K_1\tilde{m}g + K_2\omega_h + K_3p_{\omega_h} - K_4p_{\omega_\omega} \tag{47}$$

and with

$$\begin{aligned} K_1 &= (1+\beta\gamma+\delta)I + \alpha\hat{\theta}_0 \\ K_2 &= \alpha\hat{\theta}_0\tilde{p_{\omega}}_h^{\perp} + \beta\tilde{m}\tilde{g} + \tilde{p}_{\omega_h} \\ K_3 &= \gamma(I + \alpha\hat{\Theta}_0(I - \frac{gg^{\perp}}{||g||^2})) \\ K_4 &= \gamma \end{aligned} \tag{48}$$

and $I \in \Re^{3x3}$. It is possible to see that this controller stabilize the upright equilibrium.

The Implementation of the Non Linear controller can be seen below:

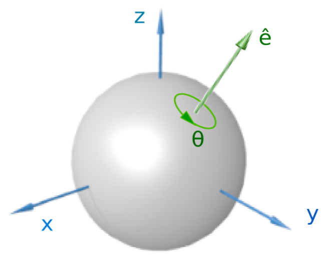
```
%-----------------------------Parameters----------------------------------
teta_0_matrix = paramCell{1};
m_tot_vector = paramCell{3};


alfa = 15;
beta = 18;
gamma = 12;
delta = 10E-5;

omega_h =(teta_0_matrix)\(p(4:6)-p(7:9));
omega_h_skew = skew3x3(omega_h);
m_skew  = skew3x3(m_tot_vector);
m_skew_nominal = skew3x3(m_tot_nominal);
g_skew = skew3x3(p(1:3));


%-----------------------------Controlled Action-----------------------------

p_omega_g = p(4:6)'*p(1:3)* p(1:3)/(norm(p(1:3))).^2;

p_omega_h = [p(4),p(5),p(6)];

p_perp_omega = p(4:6) - p_omega_g ;

k1 = (1 + beta * gamma + delta)*eye(3) + alfa*teta_0_matrix;

k2 = alfa*teta_0_matrix*skew3x3(p_perp_omega) + beta * m_skew_nominal * g_skew + skew3x3(p_omega_h);

k3 = gamma*(eye(3)+alfa*teta_0_matrix*(eye(3)-(p(1:3)*p(1:3)')/((norm(p(1:3))).^2)));

K4 = gamma * eye(3);

T = k1*m_skew_nominal*p(1:3) + k2*omega_h + k3*p(4:6) - K4*p(7:9);

%--------------------------------------------------------------------------
```

Figure 4: Controller Structure

4.4 Animation

The animation is constructed using openGL rendering that construct a moving image of a cube balancing on a corner, allowing us to see the evolution of the system showing most part of Cubli's system, excluded the spinning velocity of the reaction wheels. The animation is constructed following the following procedure: first, we costruct the cube and its axis as 3D image where initially are aligned with the inertial reference frame. Then the cube is rotated using Euler axis and angle of rotation according to the euler theorem about rotation, both extracted from its quaternions. The conversion is done considering a specific relation between quaternions and Euler axis, this allow us to move the initial figure to the right attitude. The relation can be seen here:

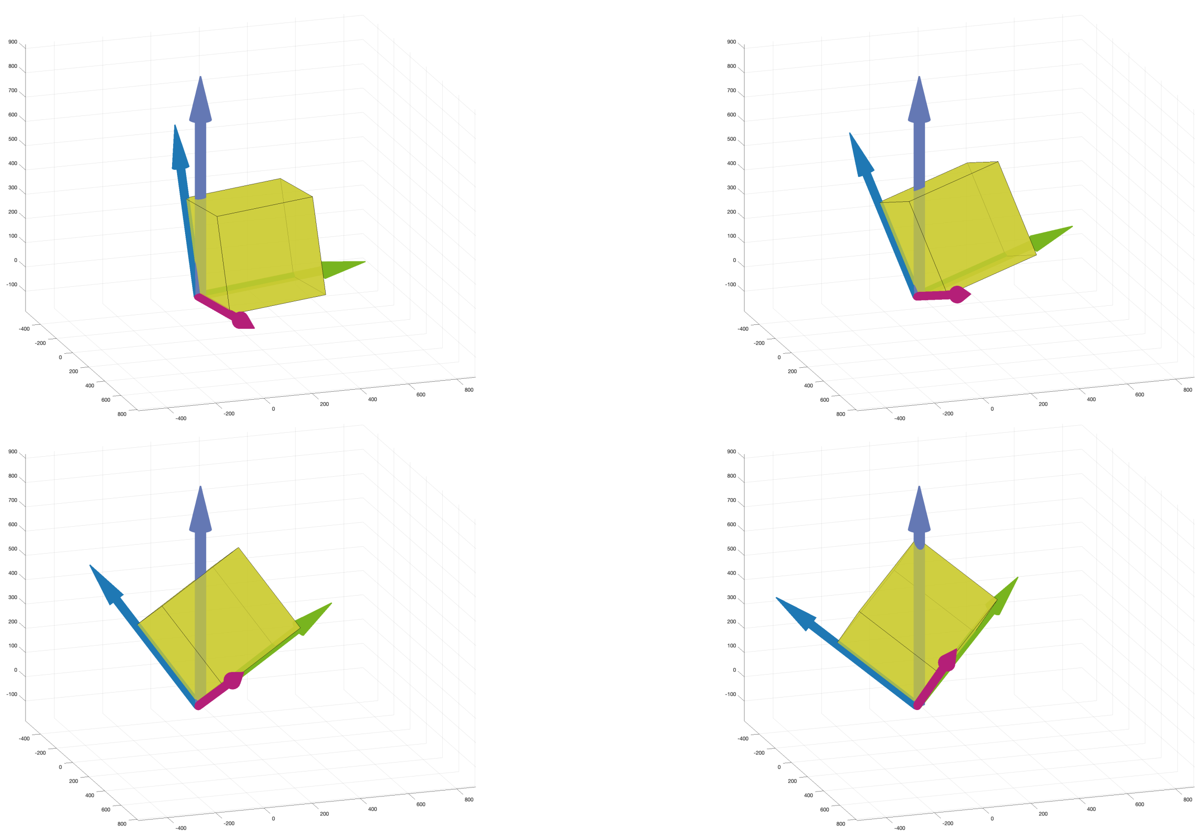
The Euler Axis $e = [e_1 \;\; e_2 \;\; e_3]$ where
$$e_1 = q_2/norm([q_2 \;\; q_3 \;\; q_4])$$
$$e_2 = q_3/norm([q_2 \;\; q_3 \;\; q_4])$$
$$e_3 = q_4/norm([q_2 \;\; q_3 \;\; q_4])$$
and $\theta = 2\cos^{-1}(q_1)$



A quaternion q is an entity that rapresents a complex number composed by 4 elements, $q = [q_0 \;\; q_1 \;\; q_2 \;\; q_3]$: three components are the vectorial part while the last one is the scalar part. We use it to calculate the homogeneous transformation from the body frame to the intertial frame. Generally, it is described by a sequence of three rotation with different conventions.

From the Euler theorem about rotations, we know that a sequence of three rotations can be represented by a single rotation around a fundamental axis called Euler Axis by an angle θ, as showed in figure above.

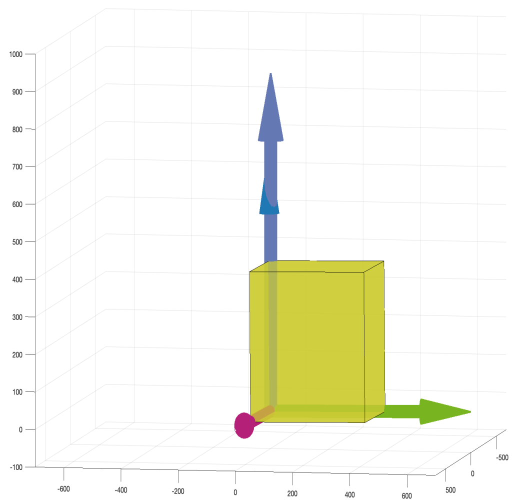
The final result of the animation is a sequence of small rotation evaluated every sampling instant.

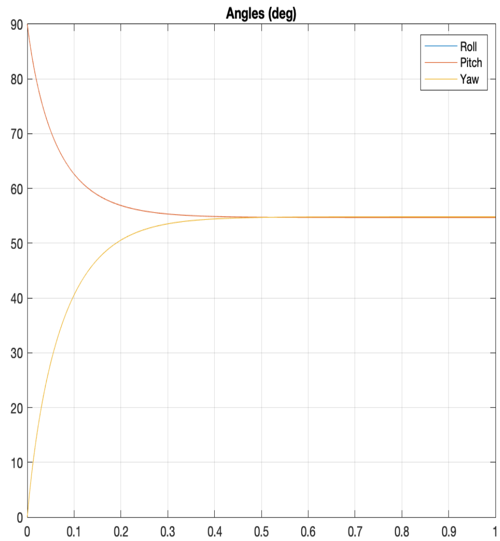


4.5 Experimental Results

Once setting up the simulator, is possible to test it in a variety of cases.

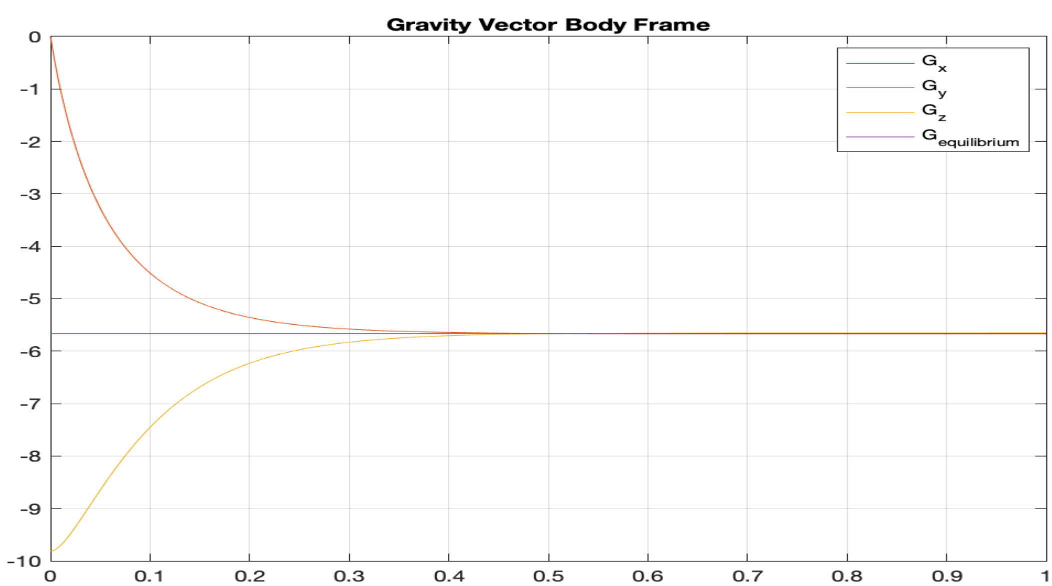
4.5.1 First scenario



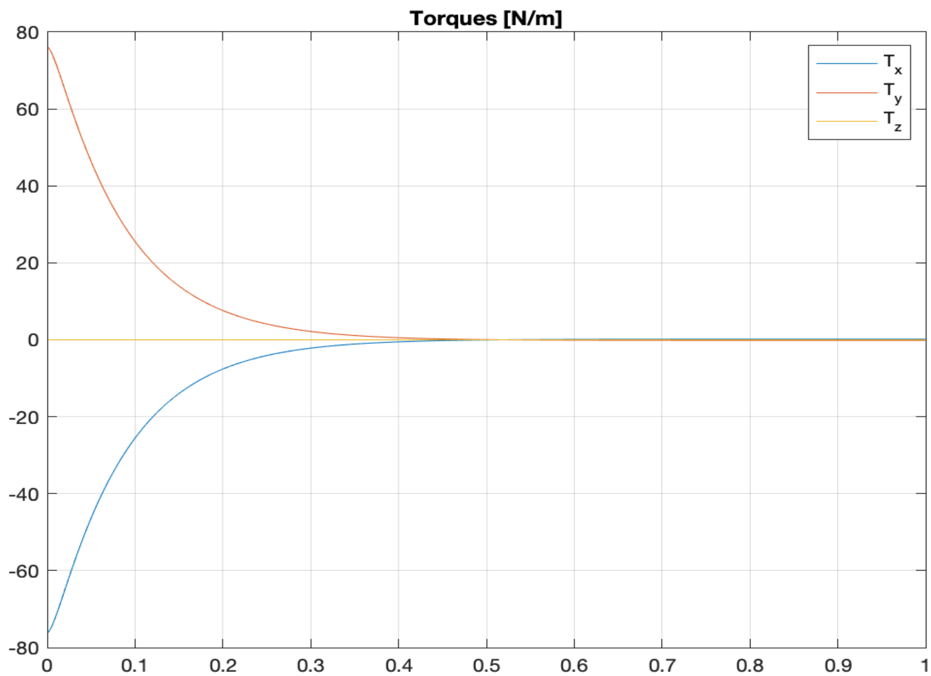


The first test showed in the image, is the one that consider the Cubli starting at rest, with angular momentum of the wheels and angular momentum at the vertex equal to 0, as it is laying on a table. The initial angular position of the body frame w.r.t. the gravity vector taken in the opposite direction - the purplue arrow in figure- is equal to $\pi/2, \pi/2, 0$. Moreover, the body frame is considered aligned to the inertial reference frame.

$(gravity\ vector\ body\ frame)$ $\quad g_0 = [0 \ \ 0 \ -9.81]$, so gravity vector point downwards in the opposite direction of our z-axes.



This image shows how the gravity vector, represented in the body frame, evolves during the maneuver of balancing on the vertex. The value reached by the gravity vector is the equilibrium point, as described in the chapter of equilibrium analysis, where the vector gravity is parallel to vector m, while the cube is balanced on a vertex. Since the vector m is the one that goes from the vertex and pass throw the center of mass, the values of the component's vector are equal to each other, indeed the final g vector has the same characteristic.

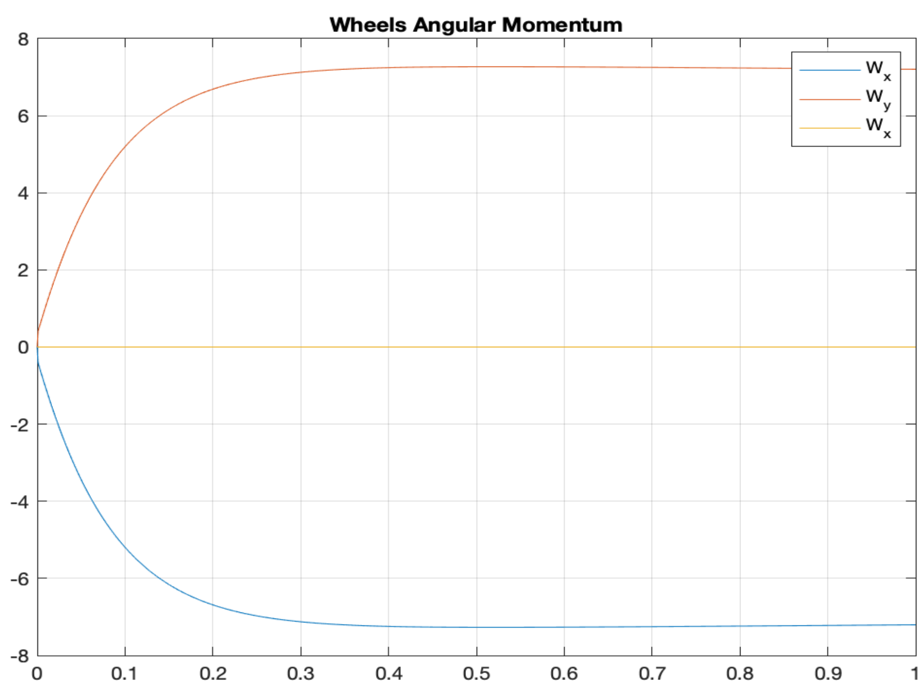


Torques evaluated by the controller are very high at the beginning because Cubli needs to move from a rest condition to arrive in the equilibrium. The more we are close to equilibrium the more torques are smaller, consequently also the opposite condition is true, so the more we are far from the equlibrium the more we need high torques. This imply that a physical implementation needs motors that are able to reach those torques, or same special tricks that allow to reach it, otherwise is not possible to reach the equilibrium. A smart trick can be bring the Cubli in a conditions that is near the equilibrium, for example using a quick breaking of reaction wheels while they are spinning at high velocity, this maneuvers generates very high torque.

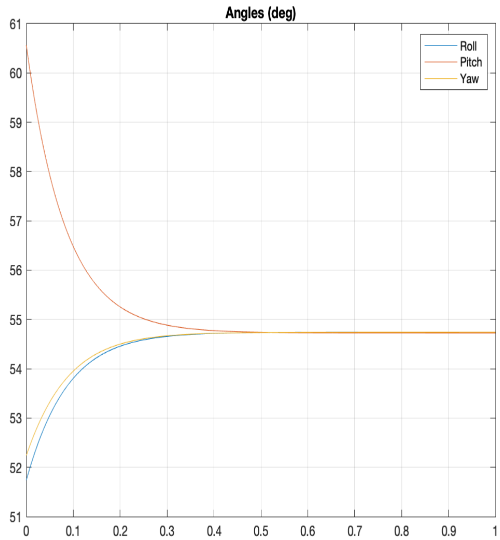
The momentum of the reactions wheels is the integral of the torque generated by the non-linear controller, indeed the controller needs to increase the wheels angular momentum in order to generate torques. At the end of the simulation, once the cube is in equlibria we have that wheels are spinning at costant velocities. Then It's possible to associate to wheels momentum the velocities of rotation of each reaction wheel by calculate it using:

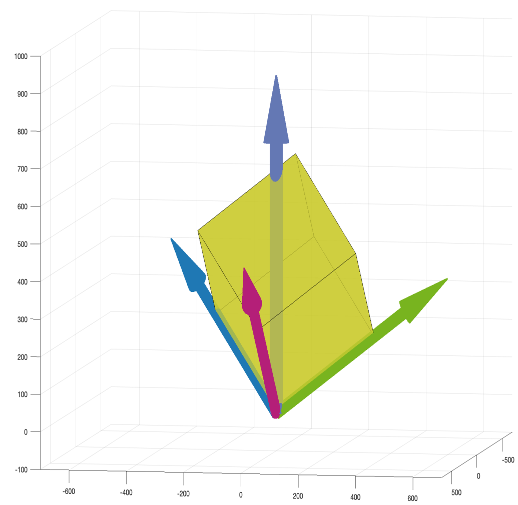
$$\omega_w = \Theta_w^{-1}(p_w) \tag{49}$$

The reaction wheels velocities is inversely proportional to inertia wheels. In our case, inertias are very small so velocities needs to be very high in order to deploy sufficient torques, as is shown in the figure below. Also in this graph is possible to notice that the action on the z-axis is equal to 0. In fact, during the motion in this particular initial condition the controller doesn't need to apply any action on this axis.



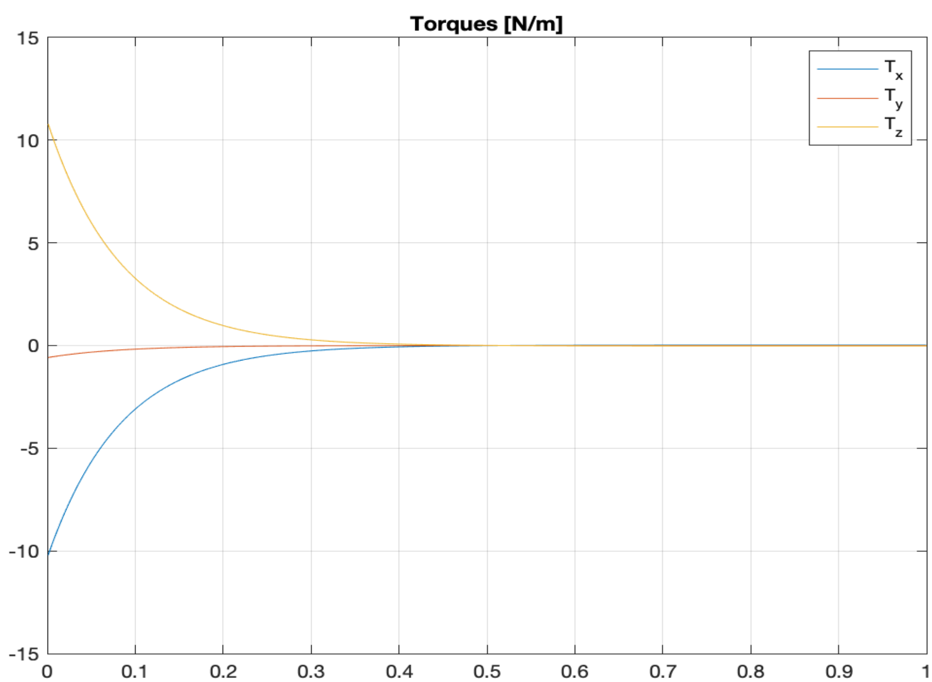
4.5.2 Second Scenario



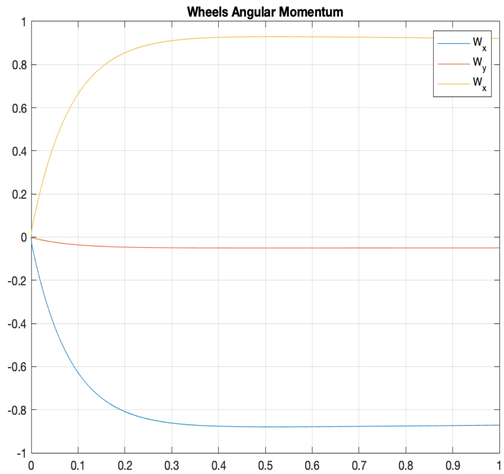


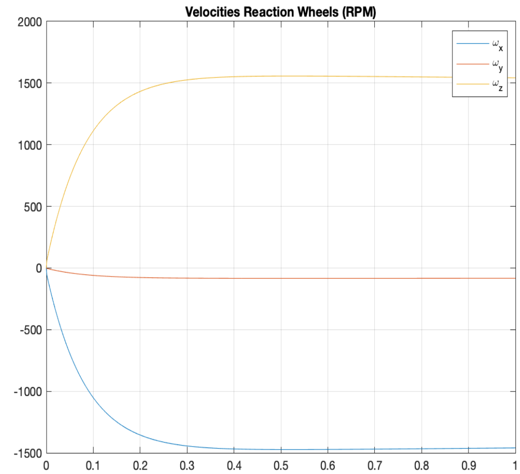
In the second scenario the Cubli starts, always, with zero angular momentum of the wheels and zero angular momentum at the vertex. However, in this case we try to verify the hypothesis we wrote before putting the cube near the point of equilibrium.

We can notice that, since the equilibrium is near the torque needed are lower than the previous case, as is showed in the figure below, this is the proof of what we was saying:



Associated to low torques we have also the reduction of the angular momentum and reaction wheels spinning velocities. As described in the previous case, the controller action is smaller if we are close to the equilibrium.

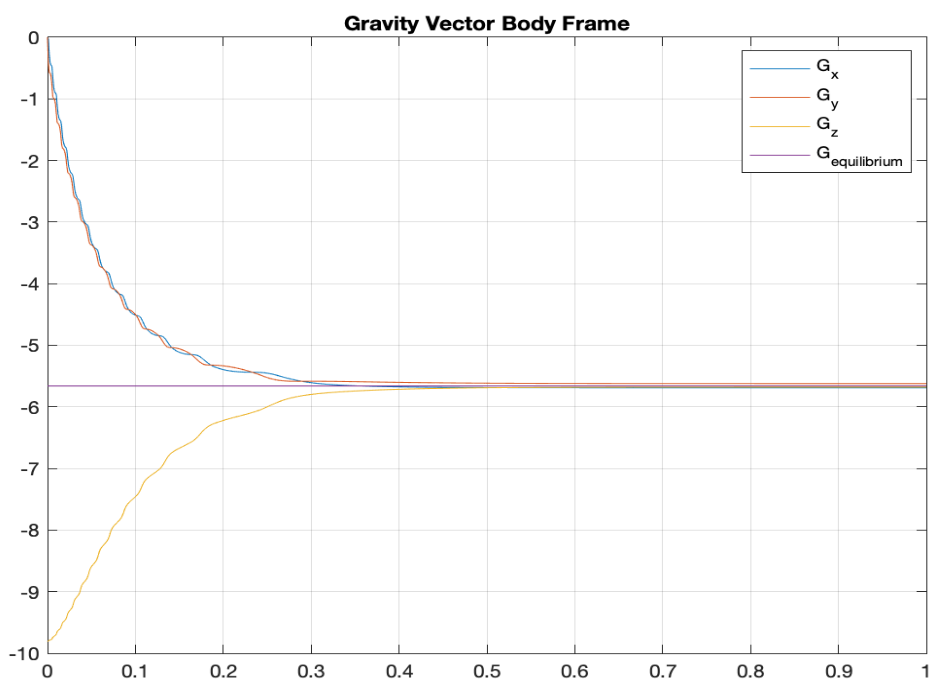




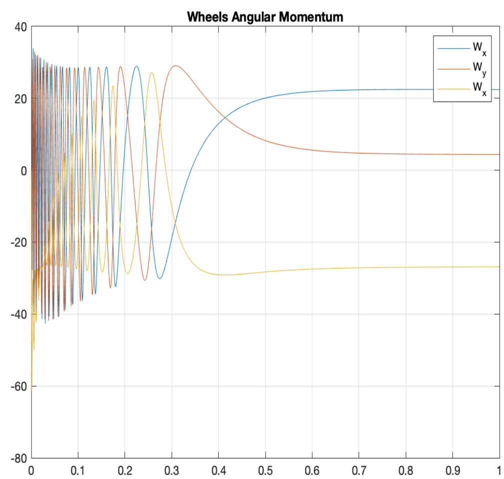
Less control imply less accumulated spinning during the balancing so it will be fast the procedure of spin reduction of the wheels.

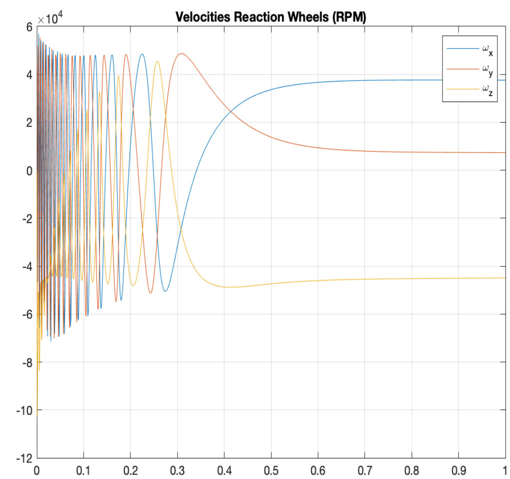
4.5.3 Third scenario

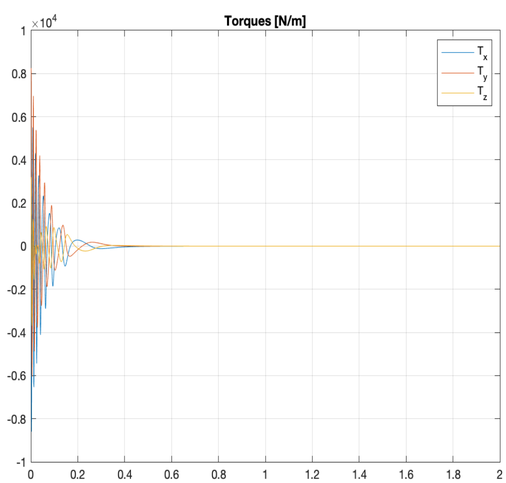
In the third scenario we add at the first case an initial high angular momentum to the vertex acting around x-axis. The exponential convergence of the g-vector position is conserved but the shape is waved, this is due to the spinning of Cubli while the controller bring it to the equlibrium.



The angular momentum and Reaction wheels velocities are very high and torques also needs to compensate the high initial angular momentum of the Cubli. Settling time, is more or less maintained, less than 1 second, from the figure is possible to see that it's about 0.7 seconds.







References

[1] Michael Muehlebach, Gajamohan Mohanarajah, and Raffaello D'Andrea. Nonlinear analysis and control of a reaction wheel-based 3D inverted pendulum. In *Conference on Decision and Control*, pages 1283–1288, 2013.

[2] Michael Muehlebach, Raffaello D'Andrea. Nonlinear Analysis and Control of a Reaction-Wheel-Based 3-D Inverted Pendulum. In *IEEE Transactions on Control Systems Technology*, pages 235–246, IEEE, 2017.