AUTOMATIC MASS BALANCING AND INERTIA IDENTIFICATION FOR A DYNAMIC CUBESAT ATTITUDE SIMULATOR

ANTON BAHU[1]

This work describes the automatic balancing and inertia identification system for a three degrees of freedom CubeSat attitude simulator testbed. For a reliable check of the attitude control algorithms, the on-orbit environment has to be simulated within the testbed. The external disturbances acting on the satellite mockup shall be minimized, the most significant one being the gravity torque. An automatic balancing procedure can largely reduce the time necessary for tuning the platform and minimizing the residual disturbance torque. The automatic balancing system adopted in this work employs three sliding masses independently actuated by three electric motors. A control algorithm automatically adjusts the location of the sliding masses to eliminate the unknown offset between the center of rotation and the center of mass. Inertia parameters are estimated through the balancing procedure by collecting free oscillating platform data. No actuators except the motors are required. Experimental results show the effectiveness of the implemented approach in achieving a good disturbance reduction.

INTRODUCTION

The growing interest for the development of highly capable nanosatellites for a wide range of missions demands for careful performance assessment of the platform through extensive ground testing. In particular, testing of the Attitude Determination and Control Subsystem is of paramount importance. To simulate the on-orbit environment, a disturbance-free rotational dynamic shall be created. A class of simulators developed specifically for CubeSats that meets such requirements are based on air bearings [4][5][6], as they offer nearly torque-free motion. The main disturbance torque affecting such kind of test benches is the one due to gravity. To reduce the gravitational torque, the distance between the center of mass and the center of rotation must be minimized.

The first approach used to reduce the gravitational torque was manual balancing, based on inspecting the spacecraft simulator pendulum motion, as described in [7]. The balance masses are moved so as to increase the pendulum period. This method requires multiple trials and is limited in real applications because of the rotational travel constraints of the spherical air bearing. Other mass balancing systems developed before [8] were based on input-output data processing of two types: based on external control moment gyros used to track the angular momentum or least-square identification based type. Most accurate results were achieved through least-squares based techniques. The overdetermined linear system of sampled data, based on inverse dynamic model or energy model, is solved. This is a dynamic procedure where a known input trajectory of control torques is fed to the system. From information obtained about the system through the dynamic or energy system model it is possible to determine the positions of three proof masses to balance the system. The limitation of least-squares-based techniques is that the estimation and compensation process must be repeated several times before accurate balancing is achieved. The drawback of the control moment gyros is the need of the external known input – the non-modeled disturbances must be neglected.

An alternative to the unknown parameters estimation is to balance the system through a feedback loop. For two axis balancing procedure this can be done by a PID based control or by more complex schemes as the one presented in [9], where the system drives the balance masses to compensate the imbalance coming from the difference in total impulse exerted by the actuator during limit cycle.

The presented work combines automatic balancing control law and inertia identification of the system to eliminate the disturbance due to the gravitational torque. The control law and identification algorithm are implemented and tested on a system under development at University of Bologna. Inspired by the work in [1], the proposed automatic balancing system uses three sliding masses controlled independently. The control algorithm automatically adjusts the location of the sliding masses to eliminate the unknown offset between the center of rotation and center of mass.

To this end, a two-step procedure is employed: the first step is devoted to in-plane balancing, the second one to the joint identification of the vertical unbalance plus the inertia properties. Since the torque that can be generated by the balance masses is physically confined in the direction perpendicular to the gravity field, the disturbance torque acting on the same subspace can be compensated by a feedback law. The compensation is achieved through a nonlinear attitude control law [2] modified to match our constrained system. In the target state subspace, the platform is spinning about the gravity vector and the attitude is forced to have the x-y body frame plane perpendicular to the gravity vector. Attitude quaternions and angular velocity values, which are required by the controller, are made available by an IMU mounted on the platform.

[1] PhD candidate, University of Bologna, Department of Industrial Engineering, Via Fontanelle 40, 47121 Forlì, Italy, anton.bahu2@unibo.it.

Once the steady state is reached, the x-y components of the unbalance vector can be computed. The feedback law is proved to be robust against disturbances and parameter uncertainties through simulations. Once the offset in the two directions perpendicular to the gravity field are known, the last component can be estimated through a least squares filter. For a straightforward least squares formulation, actuators are needed. For example, for identification and balancing of an 800 kg satellite in [8], reaction wheels are used. Different formulation can be found in [10] [11]. A survey on earlier identification techniques is done in [12].
In case a reference control input cannot be provided, previous techniques are not more applicable. Few identification techniques for inertia parameters estimation specific for the frictionless suspension of the testbed platform with lack of external actuation were developed. A possible solution for a well modeled plant is the Kalman filtering. In [3] a least square formulation was developed to estimate the CM to CR vector together with the inertia parameters for a passive system. In this case the identification is based on the sampling of free oscillating rotations.
The filter used in this work is a modification of the one documented in [3]. By sampling the free oscillating rotations, the six elements of the inertia matrix and the third offset component can be retrieved.

PLATFORM KINEMATICS AND DYNAMICS

In the modelled system the center of rotation is fixed to a point in the inertial coordinate system. The spacecraft three-axis simulator testbed structure can rotate freely but cannot translate.
The platform is approximated to a rigid body with moving point masses. The balance masses can move only along corresponding unit axis, perfectly aligned and parallel with the simulator body reference frame axes. This assumption is possible since the orientation of the simulator reference frame can be chosen freely (Fig. 1).
The rotational kinematics of a body with fixed center of rotation can be described using quaternions. Let's define absolute angular velocity $\omega = \left[\omega_x\ \omega_y\ \omega_z\right]^T$ and attitude describing quaternions vector $q = [q_1\ q_2\ q_3\ q_4]^T$. The kinematics are described by (1).

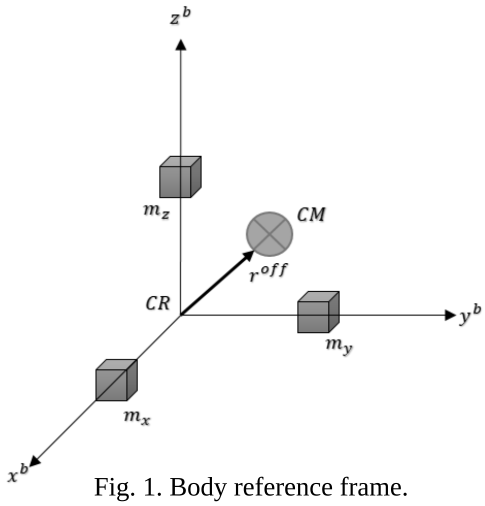

Fig. 1. Body reference frame.

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} = \frac{1}{2}\begin{bmatrix}0 & \omega_z & -\omega_y & \omega_x\\ -\omega_z & 0 & \omega_x & \omega_y\\ \omega_y & -\omega_x & 0 & \omega_z\\ -\omega_x & -\omega_y & -\omega_z & 0\end{bmatrix}\begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix} \tag{1}$$

The attitude of the body frame with respect inertial frame can be described through the quaternions (2).

$$R_i^b = \begin{bmatrix}1-2(q_2^2+q_3^2) & 2(q_1q_2-q_4q_3) & 2(q_1q_3+q_4q_2)\\ 2(q_1q_2+q_4q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_4q_1)\\ 2(q_1q_3-q_4q_2) & 2(q_1q_3+q_4q_1) & 1-2(q_1^2+q_2^2)\end{bmatrix} \tag{2}$$

Following the system model presented in [9], we assume the three balance masses move along some unit vector directions $[u_1\ u_2\ u_3]^T$, which are parallel and aligned to the axes of the body reference frame $[x^b\ \ y^b\ \ z^b]^T$.
As long as the net displacement of the balance masses can create a three-dimensional mass shift, the vector directions $[u_1\ u_2\ u_3]^T$ can be chosen freely, but the previous assumptions will simplify our model. The actual mass displacement directions can be mapped by a change of coordinate to align them with the chosen reference frame. The vectors ρ_1, ρ_2, and ρ_3 are the origins of the balance masses displacement and d_1, d_2, and d_3 the vectors describe the masses displacement with respect the zero locations. The mass positions with respect the simulator origin O_{CR} are described by (3).

$$r_i = \rho_i + d_i u_i \qquad (\, i = 1 \cdots 3) \tag{3}$$

Assume to know the total mass of the simulator including balance masses sum $m_b = m_1 + m_2 + m_3$. r_0 is the center of gravity without balance masses. The CM to CR offset vector r^{off} with respect the body reference frame is computed by (4).

$$r^{off} = \frac{1}{m}\int_B r\,\mathrm{d}m = \frac{1}{m}\left[(m-m_B)r_0 + \textstyle\sum_{i=1}^{3} m_i r_i\right] \tag{4}$$

Following new mass displacement, the offset vector changes as in (5).

$$r^{off\prime} = \frac{1}{m}\left[(m-m_B)r_0 + \textstyle\sum_{i=1}^{3} m_i\,(\rho_i + (d_i + \Delta d_i)u_i)\right] \tag{5}$$

The inertia matrix is composed of two components: the estimated simulator inertia matrix without balance masses $\hat{J}_s$ and the contribution due to the balance masses (6).

$$\hat{J} = \hat{J}_S + \textstyle\sum_{i=1}^{3}(-m_i[r_i\times][r_i\times]) \tag{6}$$

The updated inertia matrix considering masses new displacement can be computed by (7).

$$\hat{J}' = \hat{J} - \sum_{i=1}^{3}(-m_i[r_i \times][r_i \times]) + \sum_{i=1}^{3}(-m_i[r'_i \times][r'_i \times]) \quad (7)$$

Define the total moment of inertia Γ, $m_{S/C}$ the total simulator mass, r^{off} constant vector from the center of rotation to the center of mass, g the gravitational acceleration vector, the rotational dynamics of the simulator in the inertial reference frame with input torque τ_r are given by (8).

$$\dot{\Gamma} + \omega \times \Gamma = r^{off} \times m_{S/C} g^b + \tau_r \quad (8)$$

The gravitational acceleration vector g^b is expressed in the body reference frame (9).

$$g^b = R_i^b g^i \quad (9)$$

The total momentum Γ is defined in (10).

$$\Gamma = J\omega + \sum_{i=1}^{3} R_i \times m_i \dot{R}_i \quad (10)$$

The inertia matrix is time independent, the total momentum is $J\omega$ so the system dynamic equations are described by (11).

$$J\dot{\omega} + \omega \times J\omega = r^{off} \times m_{S/C} g^b + \tau_r \quad (11)$$

The only control torque provided is due to the balancing masses (12).

$$\tau_r = m_p \sum_i r_i \times g^b \quad \text{for } i = x, y, z \quad (12)$$

The control torque is constrained to lie in the direction normal to both the masses position vector and to the gravitational field direction. The controller design account for a single mass driven by a three-dimensional displacement vector, equivalent to three independent masses. The balancing masses are supposed to be equal.

After the control law is designed, the control mass displacement components r_i must be computed. It is easy to verify [9] that a displacement vector computed by:

$$r = \frac{g^b \times \tau_r}{\|g^b\|^2 m_p} z \quad (13)$$

provides the desired control torque τ_r. The compensation of two components of the offset vector can be done by an attitude feedback control law projected on the subspace orthogonal to the gravity direction, as shown in the following sections.

FULL ATTITUDE CONTROL

The offset vector from the center of rotation to the center of mass r^{off} is a three dimensional vector. An automatic balancing by a feedback nonlinear control law can be performed for two components out of three. The proposed control law is based on a projection of the full attitude control law with disturbance compensation developed in [2], which will be briefly recalled here for clarity. Let's assume the target attitude is described by $q_{ref} = [\ 0\ 0\ 0\ 1\ ]^T$ quaternion, equivalent to z^b and g^b aligned. The control law goal is to drive $q_e = [q_1\ q_2\ q_3\ ]^T$ vector to zero asymptotically.

Let's rewrite the dynamic equations in a matrix form:

$$[J]\dot{\omega} = -[\widetilde{\omega}][J]\omega + \tau_r + L + \Delta L \quad (14)$$

The L and ΔL are known and unkown distrubances respectivly. Assume the external disturbance is completely unknown and bounded. To compensate this term, an augmented state $z(t)$ is added to our control problem, designed as in (15).

$$z(t) = \int_0^t K q_4 q_e dt + [J]\omega \quad (15)$$

The augmented state of our system is $x = [q_e\ \omega\ z]^T$. Note that if the control law force $z(t)$ to remain bound, then q_e is driven to zero by design. Our system has to be asymptotically stabilized at the origin. The Lyapunov function used to design the control law is the following:

$$V(q_e, \omega, z) = \frac{1}{2}\omega^T[J]\omega + K{q_e}^T q_e + \frac{1}{2} z^T K_I z \quad (16)$$

The time derivative of the Lyapunov function along the motion of the system leads to (17)

$$\dot{V}(q_e, \omega, z) = (\omega + K_I z)^T (K q_4 q_e + [J]\dot{\omega}) \quad (17)$$

The time derivate of the quaternion error vector is provided in (18)

$$\dot{q}_e = \frac{1}{2}(q_4[I_{3x3}] + [\tilde{q}_e])\omega \quad (18)$$

Following the inverse Lyapunov theorem, Lyapunov function derivative has to be negative definite. The goal is to design the input torque so as the Lyapunov function derivative assume the following form:

$$\dot{V}(q_e, \omega, z) = -(\omega + K_I z)^T P(\omega + K_I z) \quad (19)$$

Equating the (16) and (19):

$$\dot{V}(q_e, \omega, z) = -(\omega + K_I z)^T [P](\omega + K_I z) = (\omega + K_I z)^T (K q_4 q_e + [J]\dot{\omega}) \quad (20)$$

the desired closed loop dynamic is given by (21).

$$[J]\dot{\omega} + K q_4 q_e + [P]\omega + [P]K_I z = 0 \quad (21)$$

The following control torque will stabilize our system:

$$\tau_r = -K q_4 q_e - [P]\omega - [P]K_I z + [\widetilde{\omega}]J\omega \quad (22)$$

Since Lyapunov function is negative semidefinite, the state is only Lyapunov stable; further $(\omega + K_I z)$ goes to zero. Higher order derivatives shall be inspected to check for asymptotic stability, which is indeed ensured, as shown in [2].
For a non-zero ΔL the first derivative of the Lyapunov function is not negative semidefinite and the stability of z and q_e are not guaranteed:

$$\dot{V}(q_e, \omega, z) = -(\omega + K_I z)^T (P(\omega + z^T K_I) - \Delta L) \tag{23}$$

However, for a bounded ΔL, ω and z cannot grow unbounded. For sufficiently large values of ω and z, the term

$$-(\omega + K_I z)^T P(\omega + z^T K_I) \tag{24}$$

will dominate and the Lyapunov rate will become negative. By definition, z converges to a finite value and both $\dot{\omega}$ *and* q_e must decay to zero. ω converges to zero due to the kinematic relationship.

CONTROL LAW PROJECTION

The control law presented in the previous section needs to be modified since our system is underactuated. Partial quaternion error vector compensation is made possible by a control law projection.
The angular velocity ω can be divided in two components: a projection along g^b denoted as ω_g and the complementary component ω_p representing the angular velocity perpendicular to the Earth gravitational field (26-27). The projection operator used is defined as a function of the vector g^b (25).

$$P_p(q) = \left[I - \frac{g^b (g^b)^T}{\|g^b\|^2} \right] \tag{25}$$

$$\omega_p = P_p(q)\omega \tag{26}$$

$$\omega = \omega_g + \omega_p, \quad \omega_g^T \omega_p = 0, \quad (g^b)^T \omega_p = 0 \tag{27}$$

Similarly, components of the quaternion error vector can be defined. Let's define the quaternion representation of the attitude error vector projection on the plane perpendicular to the gravity as q_{ep} and the complementary component as q_{eg} (28-29).

$$q_{ep} = P_p(q) q_e \tag{28}$$

$$q_e = q_{ep} + q_{eg}, \quad {q_{eg}}^T q_{ep} = 0 \tag{29}$$

Our target is to stabilize the platform so as z^b axis and z^i axes are aligned. In this condition the only component of the offset vector not cancelled by the balancing masses is on the z^b axis. Driving q_{ep} and ω_p to zero will lead our platform to the desired orientation. Let us consider a slightly modified with respect (16) Lyapunov function:

$$V(\omega, q_{ep}, z_p) = \frac{1}{2}\omega^T [J]\omega + K {q_{ep}}^T q_{ep} + \frac{1}{2} {z_p}^T K_I z_p \tag{30}$$

where the augmented state vector is modified so as to compensate the projected component only:

$$z_p(t) = P_p(q) z(t) \tag{31}$$

The dynamics of the q_{ep} depends only on ω_p (32).

$$\dot{q}_{ep} = \dot{q}_e - (\dot{q}_e \cdot \hat{g})\hat{g} = \frac{1}{2}(q_4 [I_{3x3}] + [\tilde{q}_e]) P_p(q)\omega = \frac{1}{2}(q_4 [I_{3x3}] + [\tilde{q}_e])\omega_P \tag{32}$$

The derivative of the quaternion error vector projection is computed in (33).:

$$\frac{dK {q_{ep}}^T q_{ep}}{dq} = 2K \frac{1}{2}\big(q_4 \omega_p + [\tilde{q}_e]\omega_p\big)^T q_{ep} = K\big(q_4 \omega_p + [\tilde{q}_e]\omega_p\big)^T \big(q_e - q_{eg}\big) =$$
$$= K q_4 {\omega_p}^T (q_{ep} + q_{eg} - q_{eg}) = K q_4 {\omega_p}^T q_{ep} \tag{33}$$

If we assume no known external torques are present, then the control law (34)

$$\tau_r = -K q_4 q_{ep} - [P]\omega_p - [P] K_I z_p + P_p(q)([\widetilde{\omega}][J]\omega) \tag{34}$$

leads to following Lyapunov function time derivative:

$$\dot{V}(\omega, q_{ep}, z_p) = (\omega + K_I z_p)^T \big(K q_4 q_{ep} + [J]\dot{\omega}\big) = \big(\omega + K_I z_p\big)^T \big(-P\omega_p - P K_I z_p - [\widetilde{\omega}]J\omega + P_p(q)([\widetilde{\omega}][J]\omega)\big)$$
$$= -\big(\omega_p + K_I z_p\big)^T [P] \big(\omega_p + K_I z_p\big) + \big(\omega + K_I z_p\big)^T \Big(-[\widetilde{\omega}]I\omega + P_p(q)([\widetilde{\omega}][J]\omega)\Big)$$
$$\approx -\big(\omega_p + K_I z_p\big)^T [P] \big(\omega_p + K_I z_p\big) \tag{35}$$

Inertia coupling term is considered to be negligible with respect the main negative defined term since it's multiplied by a quadratic term dependent on the angular speed, and the angular speed in the balancing phase is small:

$$\big(\omega + K_I z_p\big)^T \Big(-[\widetilde{\omega}]J\omega + P_p(q)([J]\omega)\Big) = \big(\omega + K_I z_p\big)^T \Big(\big([I_{3x3}] - P_p(q)\big)([\widetilde{\omega}][J]\omega)\Big) \approx 0 \tag{36}$$

Since derivative of the Lyapunov function is negative semidefinite the closed-loop control system is Lyapunov stable. Following the same reasoning done for the fully actuated system, analyzing higher order derivatives it is possible to state that $\lim_{t\to\infty} q_{ep}(t) = 0$. The system will converge to the largest invariant set Ω contained in the space defined by the Lyapunov function. Since the Lyapunov function is stable, on $\dot{V} = 0$ set it's possible to state:

$$\lim_{t\to\infty}(P(\omega_p + z_p{}^T K_I) - \Delta L) = 0 \tag{37}$$

And since ω_p is zero:

$$\lim_{t\to\infty} z_p = P^{-1}K_I^{-1}\Delta L \tag{38}$$

The internal feedback term will compensate the unknown external disturbance.

SYSTEM PARAMETERS IDENTIFICATION

After the balancing on the x-y plane is done, the offset vector r_{off} is partially known. The last component of the vector can be estimated. The employed identification technique needs no external actuation and is based on the sampling of free oscillating rotations, according to the method presented in [3]. The identification problem goal is to write the system equations in a linear form, so that a system of equations necessary for a least squares identification can be formulated (39).

$$\Phi\, x = b(\tau_{ext}) \tag{39}$$

In (39) τ_{ext} is the external torque, Φ is the observation matrix, and x is the vector of the dynamic parameters to be identified. In the case of a system with no actuation, the right-hand side of (39) is always equal to zero, which would require solving equations for the null-space of the observation matrix. Such a drawback can be avoided if the dynamic parameters are computed with respect a freely chosen point O attached to the body, different from both CM and CR. This way, the offset vector r_{off} is expressed as the sum:

$$r_{off} = r + \rho^o \tag{40}$$

Then by applying the Huygens-Steiner theorem to express the inertia J_{CR} as a function of J_o and r, the total angular momentum of the body with respect to the center of rotation can be formulated as:

$$J_{CR}\omega = J_o\omega + C(r)\omega + mB(\omega, r)\rho^o \tag{41}$$

where $B(\omega, r)$ is:

$$B(\omega, r) = \begin{bmatrix} -r_2\omega_2 - r_3\omega_3 & 2r_2\omega_1 - r_1\omega_2 & 2r_3\omega_1 - r_1\omega_3 \\ 2r_1\omega_2 - r_2\omega_1 & -r_1\omega_1 - r_3\omega_3 & 2r_3\omega_2 - r_2\omega_3 \\ 2r_1\omega_3 - r_3\omega_1 & 2r_2\omega_3 - r_3\omega_2 & -r_1\omega_1 - r_2\omega_2 \end{bmatrix} \tag{42}$$

and:

$$C(r) = m(r^T r[I_{3x3}] - rr^T) \tag{43}$$

If one defines the vector of unknown parameters to be estimated as: $[j^o\ r^o]^T = \left[J_{xx}^o\ J_{yy}^o\ J_{zz}^o\ J_{xy}^o\ J_{xz}^o\ J_{yz}^o\ r^o\right]^T$ the time derivative of the angular momentum is defined as in (44).

$$J_{CR}\dot{\omega} = \Omega(\dot{\omega})j^o + mB(\dot{\omega}, r)\rho^o + C(r)\omega \tag{44}$$

Collecting all dynamic parameters on the left-hand side of the dynamic equation, the (44) is written in the following matrix form:

$$[\Omega(\dot{\omega}) + [\widetilde{\omega}]\Omega(\omega)|m(B(\dot{\omega}, r) + [\widetilde{\omega}]B(\omega, r) + [\widetilde{g}])]\begin{bmatrix} j^o \\ r^o \end{bmatrix} = -m[\widetilde{g}]r - [\widetilde{\omega}]C(r)\omega - C(r)\dot{\omega} \tag{45}$$

where $\Omega(\dot{\omega})$ is defined as in (46)

$$\Omega(\dot{\omega}) = \begin{bmatrix} \dot{\omega}_1 & 0 & 0 & \dot{\omega}_2 & \dot{\omega}_3 & 0 \\ 0 & \dot{\omega}_2 & 0 & \dot{\omega}_1 & 0 & \dot{\omega}_3 \\ 0 & 0 & \dot{\omega}_3 & 0 & \dot{\omega}_1 & \dot{\omega}_2 \end{bmatrix} \tag{46}$$

The formulation as originally presented in [3], however requires the knowledge of the angular accelerations. Angular acceleration data computed from noisy angular speed samples could be very inaccurate. By integrating (45) a different formulation could be obtained, where angular accelerations are not more needed:

$$[\Omega(\omega) + \int[\widetilde{\omega}]\Omega(\omega)\,|m(B(\omega, r) + \int[\widetilde{\omega}]B(\omega, r) + \int[\widetilde{g}])]\begin{bmatrix} j^o \\ r^o \end{bmatrix} =$$
$$-\int m[\widetilde{g}]r - \int[\widetilde{\omega}]C(r)\omega - C(r)(\omega - \omega_0) \tag{47}$$

An efficient Least Square estimation needs good initial conditions to obtain best results. All three components of the angular speed have to be excited. Experiment time must be limited so as nonlinearities due to friction will not affect the estimation process. The data sampled from different experiments are merged together for a reliable estimation.

In the defined formulation there are more equations than unknowns, the problem is solved computing the LS solution of:

$$x = \Phi^{+} b(\tau_{ext}) \tag{48}$$

where Φ^+ is the pseudo-inverse of the observation matrix. This LS solution is an estimation of the vector of parameters x. The Huygens-Steiner theorem can finally be applied to find the inertia matrix of the body with respect the CM:

$$J_{CG} = J_o - m(\rho^{o^T} \rho^o [I_{3x3}] - \rho^o \rho^{o^T}) \quad (49)$$

The last offset vector is computed by (50).

$$r_{off} = r^z + \rho^{oz} \quad (50)$$

SIMULATION RESULTS

The balancing procedure efficiency was checked through the simulations using a model developed in Matlab Simulink. The results presented here are based on data estimated from the experimental data.

Since in (34) some terms of the control law foresee the knowledge of the inertia matrix, in the simulations and experiments a simplified version of the control law was used (51)

$$\tau_r = -Kq_4 q_{ep} - [P]\omega_p - [P]K_I z_p \quad (51)$$

The neglected term $P_p(q)([\tilde{\omega}][J]\omega)$ depends on the unknown inertia matrix. Since in affects only the transient and not the error at steady state it can be neglected. By comparing the results of simulations obtained by using full and reduced control laws, no stability of the reduced control is checked. In the following the reduced control law is always used.

Total mass of the platform used for the simulations is $6.3\ kg$. The masses used for the balancing are $m_x = m_y = m_z = 0.465\ kg$. The inertia matrix used is the following:

$$J_{CR} = \begin{bmatrix} 0.0241 & -0.001 & 0 \\ -0.001 & 0.0332 & -0.004 \\ 0 & -0.004 & 0.0363 \end{bmatrix} kg/m^2$$

The initial angular speed used for the plane balancing simulation is $\omega = [-0.307 \quad 0.189 \quad -1.500]$ rad/s. The sampling time is $0.05\ s$, which was found to be optimal based on our hardware.

The behavior of the vector of balancing masses positions is shown in the Fig. 2. The steady state is reached after 300 seconds. Following the defined control scheme, the only angular velocity component different from zero is aligned with z^i axis of the inertial reference frame. The g_{bx} and g_{by} components converge to zero in steady state. This behavior is in accordance with the control law analysis. The convergence of the estimated parameters is shown in the Fig. 3. It is possible to see that the only component last to be compensated after the transient is along the z^b axis, which is aligned with the z^i axis at the steady state.

The residual error of x and y components of the unbalance vector after 300 seconds is in the order of $10^{-7}m$, less than the residual disturbance expected by our hardware.

The estimation of the unknown unbalance vector component and inertia matrix is done based on data collected from a 200 seconds long simulation of free oscillation of the platform with sampling time of $0.05\ s$. The experiment time has to be limited so as the friction do not affect the collected data. Friction due to the air bearing, measurement noise on angular speed and attitude are simulated based on the datasheets of the hardware. The noisy angular acceleration computed from the angular speed measurements data is too inaccurate to be used directly in the estimation process. To verify the efficiency of estimation algorithm, the results of estimations based on the ideal angular acceleration, available in the simulation, and data estimated by the integration of the system equation are compared.

The requirement on the ρ^o vector is to be different from zero, but for a reliable estimation the magnitude should in order of the offset vector. In the simulations, the ρ^o vector is chosen to be half of the unbalance vector. The estimated data are computed with respect the R_B reference frame.

The true vector of parameters to be estimated is:

$$\begin{bmatrix} j^o \\ r^o \end{bmatrix} = x_{true} = [0.0241\ 0.0333\ 0.0364 - 0.0011\ 0.0009 - 0.0040\ 0.0004\ 0.0007 - 0.0026]^T$$

The first two components of the r^{off} vector, which are known from the plane balancing, are used as constraints in the estimation problem.

From data collected during the simulations, the vector estimated based on real angular acceleration data is:

$$x_{est} = [0.0241\ 0.0342\ 0.0371 - 0.0012\ 0.0008 - 0.0041\ \ 0.0004\ \ 0.0007\ \ - 0.0026]^T$$

The vector estimated based on the integrated data is the following:

$$x_{estan} = [0.0242\ 0.0341\ 0.0375 - 0.0012\ 0.0008 - 0.0042\ \ 0.0004\ \ 0.0007\ \ - 0.0026]^T$$

As it's possible to see from the provided data, the difference between the two estimation is negligible. The percentage error of the estimated parameter with respect the real data is the following:

$$err_{\%} = [-0.4149 \quad -0.9021 \quad -1.0220 \quad -1.0909 \quad 1.2222 \quad -1.5200 \quad 0 \quad 0 \quad -0.0769]^T$$

By iterating the experiments, the estimation error is reduced till no more appreciable variation on estimated parameters could be noticed. Due to the simulated source of errors, such as measurement noise, air friction, residual error from the first step of the balancing, the estimation error limit was found to be of 2% in the worst case.

The main source of error is the estimation of products of inertia since their magnitude is small compared with main inertia terms.

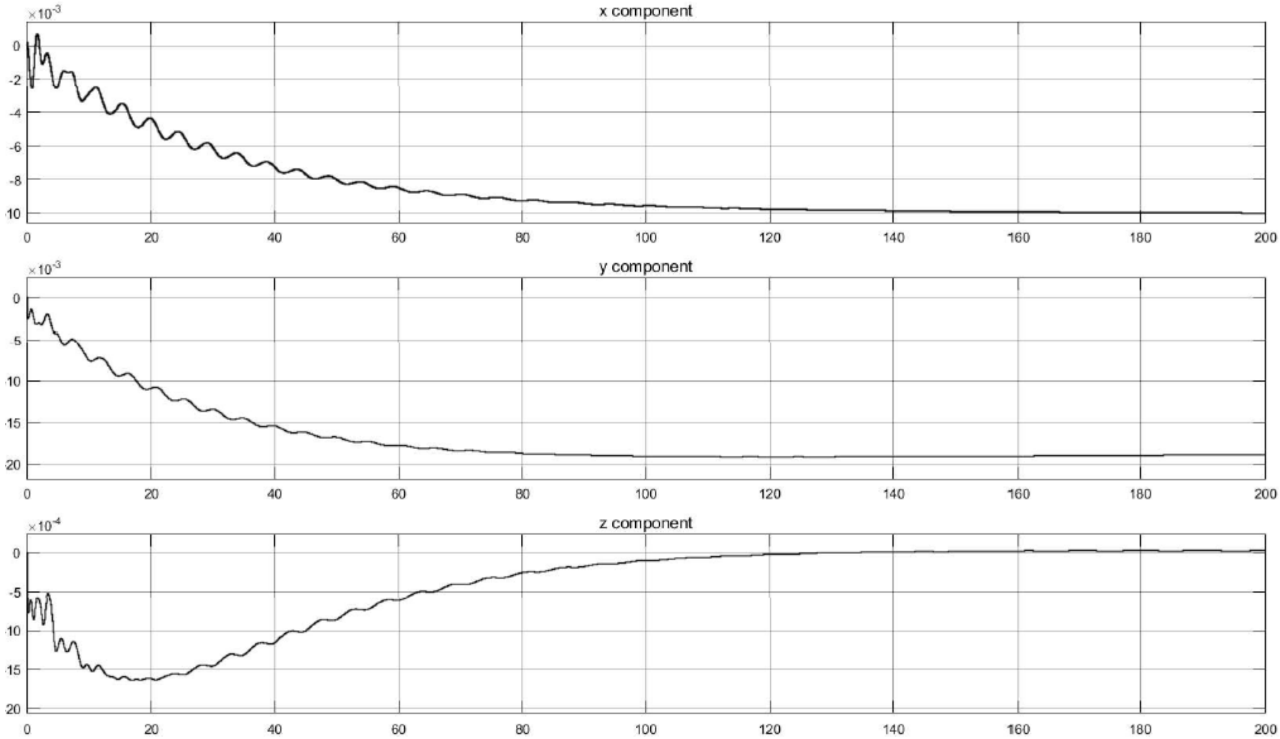

Fig.2 Balancing vector behavior - simulation

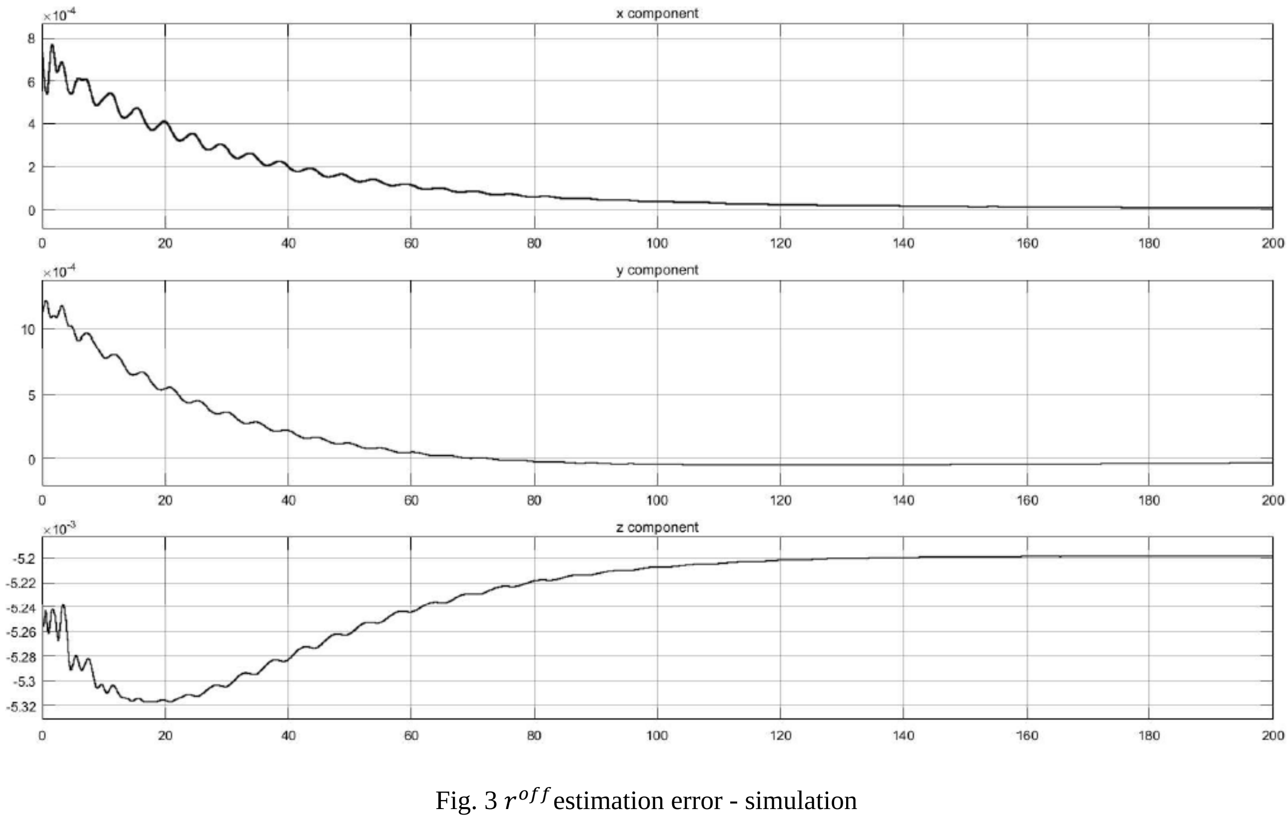

Fig. 3 r^{off} estimation error - simulation

To verify the effectiveness of the balancing procedure proposed here, the variation of the kinetic energy of the system is analyzed. The total energy of system, described by (52), is constant in case no dissipative forces are applied to our system.

$$E_{tot} = E_{kin}(t) + P_g(t) = const \tag{52}$$

$P_g(t)$ is the potential energy of the system and $E_{kin}(t)$ is the kinetic energy of the system. The total kinetic energy of the proposed model is described by the kinetic rotational energy

$$E_{kin}(t) = \omega(t)^T J \omega(t) \tag{53}$$

If the system is balanced $P_g(t) \approx 0$. Therefore, the rotational kinetic energy is constant during the motion (54)

$$E_{tot} = E_{kin}(t) = \text{const} \tag{54}$$

By analyzing the kinetic energy variation in time, the effectiveness of the balancing procedure can be evaluated. Because the rotational kinetic energy can be evaluated by the gyroscope measurement, this quantity provides an easy way to demonstrate the accuracy of the balance procedure experimentally.

The variation of kinetic before and after the balancing neglecting the air friction are shown in the Fig. 4 and 5 respectively.

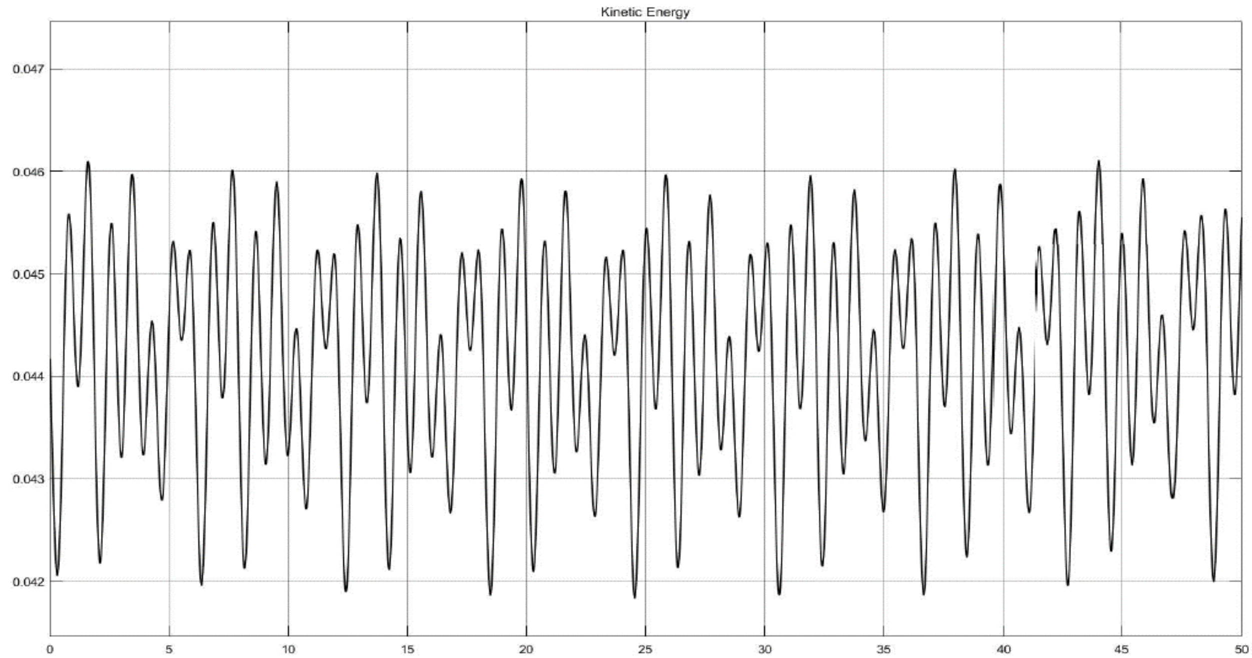

Fig. 4 Kinetic energy of unbalanced platform

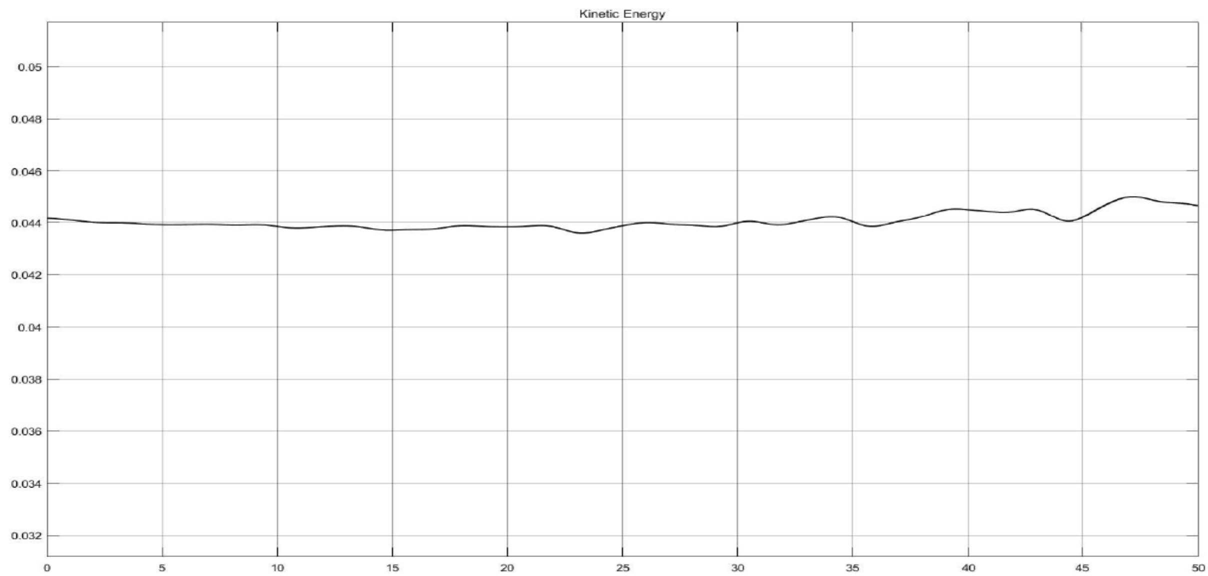

Fig. 5 Kinetic energy of balanced platform

The deviation of the kinetic energy before and after the balancing is $\sigma = 0.0011$ and $\sigma = 2.7142 \cdot 10^{-6}$ respectively, meaning a reduction of 99.75 %. In case the air friction is modelled, the kinetic energy has to be evaluated in relatively short time. The residual disturbance is due to the error in the estimation process (measurement noisy, sampling delay, air friction torque) and the modelled uncertain on motor positions.

EXPERIMENTAL SETUP

The developed algorithm was tested on the facility developed at the Aerospace Faculty of University of Bologna. The testbed provides three rotational degree of freedom with guaranteed movements of 360º in Yaw and ±50º in Roll and Pitch.
The balancing system is mounted on the platform. For this reason, all the components have to be carefully chosen to comply with the system dimension and weight requirements. The core of our system is made of Arduino components. The actuation is done by three stepper motors.
The on-board controller is an Arduino Due. The IMU is the Arduino 9-Axes Shield unit, based on Bosch BNO055 unit, implements a 9-axis Absolute Orientation Sensor. The communication is done through Arduino Wi-Fi shield.

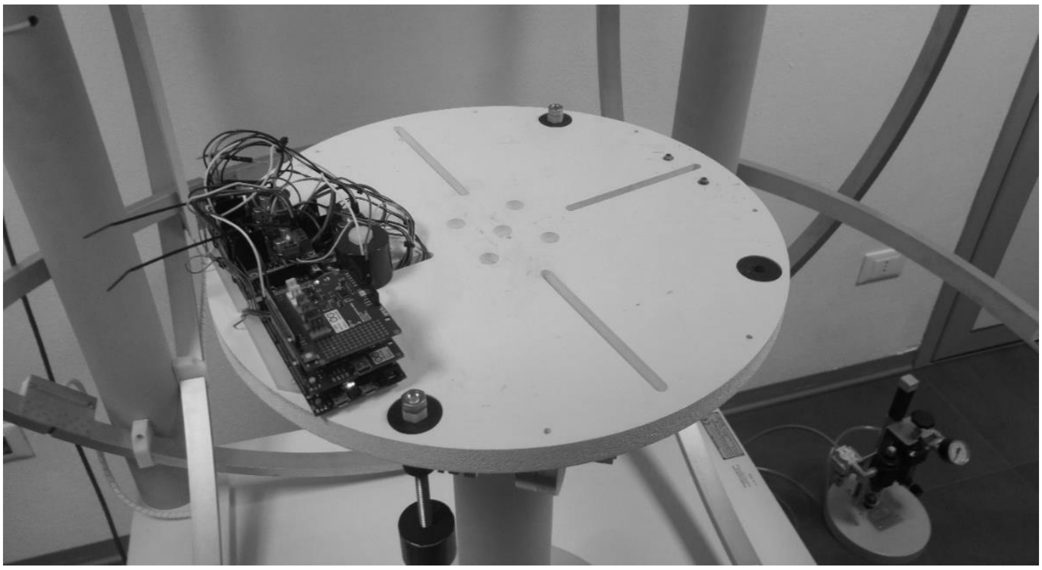

Fig. 6 Testbed experimental setup

The balance masses weight influences the maximum offset between the center of rotation and center of mass which is possible to compensate. The CubeSat Design specifications (13) limit the possible offset of a CubeSat to ± 20 mm on each of three body axes. The balancing mass for each axis can be calculated as

$$\text{mass}_{\text{balance}}(\text{single axis}) = \frac{\text{travel}_{\text{balance}}}{\text{travel}_{\text{offset}}} * \text{mass}_{\text{total}} \quad (55)$$

The motor step is 0.01 mm and the maximum disturbance allowed is of 0.001 Nm. Due to the mechanical uncertainty and backlash, the motor accuracy considered in the calculation is of 0.02 mm. The steppers motors are of non-captive type. Considering the minimum length of the axis inside the motor and space needed to fix the axis to mass itself the available travel is 74 mm (± 37 mm). The maximum offset to be compensated by the automatic balancing was chosen to be ± 2 mm, roughly one tenth of the CubeSat specifications – within this range the balancing has to be done by fixed weights and design.
The balancing mass and axis together weights 465 grams. The total mass of the platform, balance masses and the used CubeSat mockup installed is of 6300 grams. The maximum offset which can be compensated can be calculated respectively.

$$\text{Offset}(\text{single axis}) = \frac{\text{mass}_{\text{balance}}}{\text{mass}_{\text{total}}} * \text{travel}_{\text{offset}} = 2.7 \text{ mm} \quad (56)$$

The designed offset is enough to cover the target specification.
The actual balance mass positions are computed by a rotation matrix which relates balance masses reference frame and the body reference frame. In the first stage a plane balancing is performed. Since the balancing motors are not aligned with the IMU axes, the measurement data must be accounted through a rotation matrix. According to the mechanical design, 45° rotation around the z axis can be considered for the balance mass displacement computation. All the data provided are expressed in the body reference frame, supposed to be coincident with the IMU reference frame.

EXPERIMENTAL RESULTS

The initial angular speed for the plane balancing experiment is of $\omega = [0.3676 \;\; 0.1265 - 1.4835]^T$ rad/s. The sampling frequency is 20 Hz. The steady state is reached after 300 seconds. Due to highly nonlinear motor transfer function and the measurement noise affecting the integration, the transient of the balancing vector components differs significantly from the simulation data, but the general behavior is similar. The transient is shown in Fig. 7. The system is considered balanced once the components of the gravitational vector on x-y plane are in the order of the measurement noise.

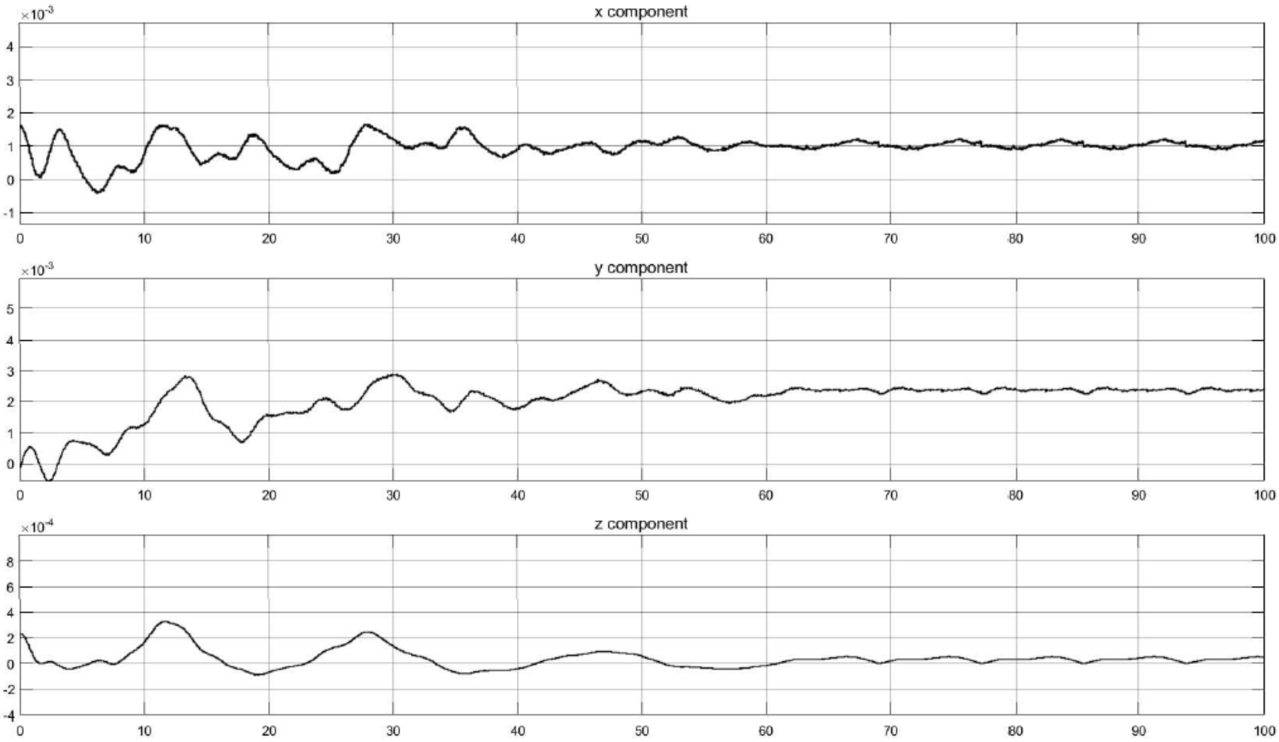

Fig.6 Masses position vector variation during the balancing experiment.

Next, a known unbalance vector is introduced. The same sampling frequency is used to collect the data for the least squares estimation. The experiments are iterated until the data are consistent with respect the previous identification. The inertia matrix found by the estimation is the following (used in the previously provided simulation data):

$$J_{CR} = \begin{bmatrix} 0.0241 & -0.001 & 0 \\ -0.001 & 0.0332 & -0.004 \\ 0 & -0.004 & 0.0363 \end{bmatrix} \text{kg/m}^2$$

The balanced platform oscillation data are sampled to verify the efficiency of the balancing procedure. Data are collected through two experiments of 40 seconds each to estimate unbalanced platform kinetic energy and balanced platform kinetic energy. The initial angular speed are respectively $\omega_{unbalanced}$ =[-0.5171 0.0742 -1.7770] rad/s and $\omega_{balanced}$=[0.3076 0.1898 -1.5010] rad/s. Since the modules of the two is close, efficiency of the balancing can be evaluated.

The disturbance moment due to the residual gravitational torque is difficult to be decoupled from the aerodynamic drag and air bearing friction. The effectiveness of the balancing is evaluated by computing the kinetic energy standard deviation, with 71% reduction, with respect the 99.75% found through the simulations. As expected, the results of the experiments are worse than the theoretical results. This difference is double-natured. Neglected disturbances such as aerodynamic drag – expected to be low due to the low angular speed module - and some mechanical details not taken in account exaggerate the performance computed by the simulations. On the other side, the balancing is not perfect: the first stage performance is degraded due the imperfect components estimation and actuation non-linearity, the second stage estimation depends heavily on the system parameters knowledge and first stage performance. Moreover, the method used to estimate the balancing efficiency, kinetic energy evaluation, is based on the knowledge of the inertia matrix, which is itself estimated through the balancing procedure and is affected by the estimation error.

CONCLUSIONS

In this work a two-step procedure to reduce the gravitational torque acting on the experimental platforms based on air bearing is presented and experimentally tested. The procedure is based on two stages: in the first stage in-plane balancing is done, in the second one vertical unbalance plus the inertia properties are estimated. A nonlinear feedback control law used in the first stage guarantees asymptotical stability around the equilibrium point, as showed through the Lyapunov function stability analysis. Once the plane balancing is done, two components of the unbalance vector are computed and used as constraints in the second stage.

In the second stage dynamic equations of the system are rewritten such as to formulate linear constrained least squares problem. Free oscillating system data are used to estimate the unknown parameters. The information necessary to balance the platform is obtained, together with an estimate of the inertia matrix.

The proposed procedure can be used on low-cost systems, since no expensive actuators for a precise control torque reference input are needed. At the same time, the system is robust thanks to a feedback law implementation.
The entire procedure sketched above was verified both through numerical simulations and experimental tests on the test-bench using some ad-hoc hardware based on the Arduino microcontroller and other low-cost components. Main nonlinearities and disturbances are modelled, and robustness of system with respect thus and the system uncertainties are proved through the simulations. Asymptotic stability of the feedback law and efficiency of the estimation are checked. Experimental results confirm that adequate performance can be reached, with the automatic balancing being effective in reducing the disturbance torque down to the hardware-dependent limit.
The implemented automatic balancing system is simple from the hardware point of view, since it needs only three actuated masses, and proves to be reliable also when low-end COTS components are used to collect and process the data. As such, it represents a viable solution for the pre-flight verification of nanosatellite attitude control systems in ground-based simulators.
The balancing performance can be improved. Regarding the first stage, better hardware, especially more precise IMU and dedicated motor drives, would guarantee a better plane balancing. More accurate mechanical design, assembling and modelling would provide a better initial compensation and better knowledge of the system parameters in the second stage.

REFERENCES

[1] Chesi, S., Gong, Q., Pellegrini, V., Cristi, R. Romano, M.: Automatic Mass Balancing of a Spacecraft Three-axis Simulator: Analysis and Experimentation. J. of Guidance, Control, and Dynamics, 37 (1), 2013.
[2] Hanspter Schaub, John L. Junkis, Analytical Mechanics of Space Systems, Second Edition. Chap. 8.5 pag. 396-402
[3] Gavrilovich, I., Krut, S., Gouttefarde, M., Pierrot, F., Dusseau, L. “Robotic Test Bench for CubeSat Ground Testing: Concept and Satellite Dynamic Parameter Identification” IROS: Intelligent Robots and Systems, 2015, Germany.
[4] Bevilacqua, D., and Gallardo, R., Six degrees of freedom experimental platform for testing autonomous satellites operations. 8th International ESA conference on Guidance, Navigation and Control Systems, 2011.
[5] T. Ustrzycki, R. Lee and H. Chesser., Spherical air bearing attitude control simulator for nanosatellites. AIAA Modelling and Simulation Technologies Conference, 2011, Portland, Oregon.
[6] J. L. Schwartz, The distributed spacecraft attitude control system simulator: from design concept to decentralized control. PhD thesis, Blacksburg, Virginia: s.n., 2004.
[7] Fullmer, R. R., Dynamic Ground Testing of the Skipper Attitude Control System, : 34th AIAA Aerospace Science Meeting and Exhibit, AIAA Paper, Jan. 1996, Vol. 1996-0103
[8] Kim, J.J. and Agrawal, B.N., Automatic mass balancing of air-bearing based three-axis rotational spacecraft simulator. Agrawal, J. Kim and B. s.l.: Journal of Guidance, Control, and Dynamics, May-June 2009, Vol. vol. 32, no. 3, pp. 1005-1017.
[9] Hatcher, N. M., and Young, R. N., Automatic Balancing System for Use on Frictionlessly Supported Attitude-Controlled Test Platforms. NASA TN-D-4426, Langley Research Center, March 1968.
[10] M. A. Peck and A. R. Cavender, An airbearing-based testbed for momentum-control systems and spacecraft line of sight. Advances in the Astronautical Sciences (American Astronautical Society), Vol. pp. 427-446
[11] ByungMoon Kim, Efstathios Velenis, Patrick Kriengsiri, and Panagiotis Tsiotras, A Spacecraft Simulator for Research and Education. Georgia Institute of Technology, Atlanta, GA 30332-0150, USA
[12] Jana L. Schwartz and Christopher D. Hall, Comparison of System Identification Techniques for a Spherical Air-bearing Spacecraft Simulator.
[13] SLO, CalPoly. The CubeSat Program, CubeSat Design Specification Rev.13. 2014.