Determining mass properties with a trifilar pendulum

Timo Horstschäfer

***Abstract*—Mass properties of rigid bodies, which include the *center of gravity* as well as the *inertia tensor* play an important role in the analysis of the motion and the behaviour of moving objects. This report describes a method of determining the stated properties by using a trifilar pendulum. After verifying the basic measurement processes, and testing it against known values, the methods are put together to obtain a three-dimensional center of gravity and the complete inertia tensor given in the center of mass frame of the object being analyzed. All steps are then carried out on the *Muscat FUU*.**

I. Introduction

THIS report describes a method of determining mass properties of an object by using a trifilar pendulum. It gives a method of measuring the three dimensional center of gravity (*CoG*) as well as the complete inertia tensor in the center of mass frame of the object being measured. These properties are necessary to determine the motion of a rigid body, e. g. to compute there behaviour as a gyroscope. This is especially important for the *Space and Plasma Physics* group at *KTH, Stockholm*, which has been sending several units on the rocktes of the *REXUS* (*Rocket Experiments for University Students*) programme. These rockets are spin-stabilized, thus carried units gyrate quickly upon ejection in the atmosphere. To analyze the measurements of these experiments, good knowledge of the mass properties is often necessary.

First, the method of measuring the two-dimensional center of mass is being described. The precision is tested with a setup of test objects of known mass properties. Next, the procedure of measuring the moment of inertia is introduced and again, tested for precision with well-defined test objects. Finally, the complete tensor of inertia, based on the previous measurement techniques is being determined for the *Muscat FFU* (*Free-Falling Unit* of the *MUltiple Spheres for Characterization of Atmospheric Temperature* experiment).

The work of this report is based on the platform built by *Ernest C. Vallet* and uses much of his previous work found in [2].

II. Center of Gravity

In a system of particles the *center of gravity* can be given by the relation [1]

$$\boldsymbol{R} = \frac{1}{M}\sum_i m_i \boldsymbol{r}_i \tag{1}$$

$$M = \sum_i m_i \tag{2}$$

We can use this relation, since we treat the platform and the unit as single particles, i. e. we have

$$\boldsymbol{R} = \frac{m_\mathrm{p}\boldsymbol{r}_\mathrm{p} + m_\mathrm{u}\boldsymbol{r}_\mathrm{u}}{m_\mathrm{p} + m_\mathrm{u}} \tag{3}$$

However, we do not know any of the coordinates $\boldsymbol{r}_i$. But since we are only interested in the position of the center of gravity of the unit, we can perform two measurements of the platform, with and without the unit, to extract the wanted information.

The measurement works by placing the platform on *three* support points, given by sharp spikes. The central spike is in the geometric center of the platform, and hence has the coordinate $\boldsymbol{s}_\mathrm{c} = (0, 0)$. The other support points are attached to the end of each arm and placed on a precision scale. When placed on the spikes, the platform stands on three support points, which are the central spike and two arms, while one arm is always free in the air (figure 2).

As the scales give the equivalent of a mass placed at the support point and the mass on all support points together sum up to M, we can write

$$\boldsymbol{R} = \frac{(M - g_1 - g_2)\boldsymbol{s}_\mathrm{c} + g_1\boldsymbol{s}_1 + g_2\boldsymbol{s}_2}{M} \tag{4}$$

$$= \frac{g_1\boldsymbol{s}_1 + g_2\boldsymbol{s}_2}{M} \qquad \boldsymbol{s}_\mathrm{c} = (0,0) \tag{5}$$

where g_i denote the values measured by the scales.

Equating with (3) gives

$$m_\mathrm{p}\boldsymbol{r}_\mathrm{p} + m_\mathrm{u}\boldsymbol{r}_\mathrm{u} = g_1\boldsymbol{s}_1 + g_2\boldsymbol{s}_2 \tag{6}$$

Repeating the measurement without the unit then gives

$$m_\mathrm{p}\boldsymbol{r}_\mathrm{p} = g_1'\boldsymbol{s}_1 + g_2'\boldsymbol{s}_2 \tag{7}$$

And substracting the two equations hence yields the *2D CoG* formula

$$\boldsymbol{r}_\mathrm{u} = \frac{(g_1 - g_1')\boldsymbol{s}_1 + (g_2 - g_2')\boldsymbol{s}_2}{m_\mathrm{u}} \tag{8}$$

where g is the *total* g value and g' is the *platform* g value. Hence, the difference $g - g'$ then represents the bare *unit* g value as wanted. The test mass can then be used to repeat this

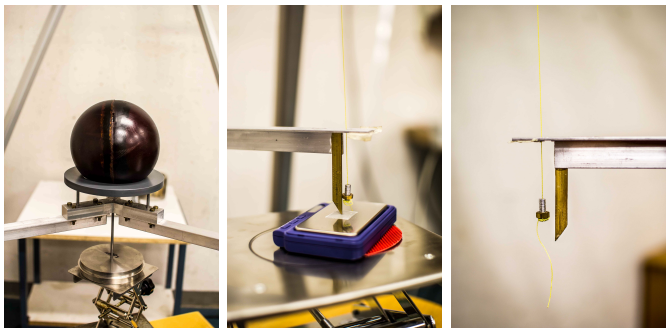

Fig. 2: Fixed platform with *Muscat FFU* on the central spike (a), support point on a scale (b) and the free arm (c).

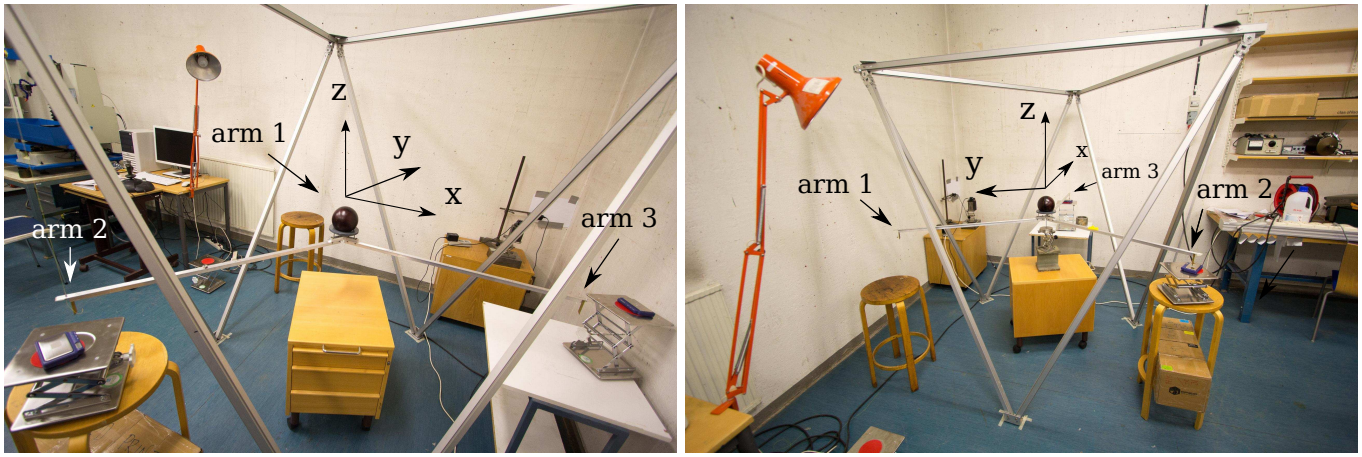

Fig. 1: Two pictures showing the trifilar pendulum with the *Muscat FFU* mounted on the plastic ring, also giving the coordinate system used for the measurements. The first picture shows the free pendulum, whereas the second shows the pendulum on the support points.

measurement under different conditions. This gives statistics about the real center of gravity of the unit and since the mass and position of the test mass is known, it adds a known contribution to the values of the scales.

A. Testing the single scale measurement

To check, if the above procedure works as expected, we need to test it against a known property. Therefore, we select one arm to be tested and perform measurements using a test mass placed at different positions on the arm, see picture 3.
Assuming, that the test weight only affects the arm, which it is placed on, thus giving $g_2 = g_2'$, we find from (8)

$$\boldsymbol{r} = \frac{(g - g')\boldsymbol{s}}{m} \tag{9}$$

Taking the absolute value, we thus have

$$g = \frac{mr}{s} + g' \tag{10}$$

$$= g_{\mathrm{t}}(r) + g' \tag{11}$$

which predicts the result of the measurements to be a linear function in r shifted by the constant value g_1'.
A test has then been performed on each arm with a nut of mass $10.40\,\mathrm{g}$ placed at positions $r \in [25, 75]$ cm in steps of 5 cm. To estimate the constant g', we take the mean difference between the measured g and g_{t}. This quantiy is used again in many other plots and is thus given as

$$\Delta g \equiv \overline{g - g_{\mathrm{t}}} \tag{12}$$

Adding it to g_{t} places the theoretical curve directly upon the measured, thus making it easy to see any deviations. The results are given in figure 4 show a very good agreement with the theoretical prediction. This procedure of using a test mass to obtain a series of g measurement is used again in the next section.

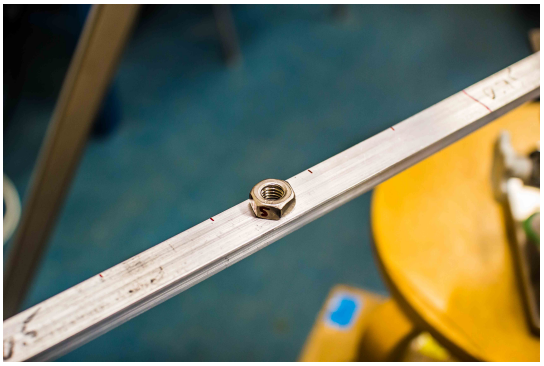

Fig. 3: A nut as test mass placed on an arm.

B. 2D center of gravity in a plane

The procedure of the section above was the extended to measure the center of gravity of a unit in the two-dimensional xy plane of the platform. Using the estimate of g' from the previous section, we could measure the value g with the unit and then find $g - g'$. However, it was found, that it is very difficult to set up the platform exactly in the same way, so that the measured value g' remains constant. Thus, to determine the center of gravity, we simply take one series g with the unit and one without. Substracting the resulting curves then directly gives us a series of measurements of $g - g'$, of which the mean value can be taken as the definite result of $g - g'$.
To test the precision of this procedure, three weights have been placed on the platform. Their weights and positions are given in table I. The position given in cm refers to the distance of the weight to the center of the platform. Since all positions and masses are known, the resulting center of gravity can be directly calculated by using equation (2). The result is given as $\boldsymbol{R}_{\mathrm{t}}$ in table III. Five positions of the test mass ranging from 25–75 cm in steps of 10 cm have been used to determine $g - g'$. As a constant quality control, the curve $g_{\mathrm{t}} + \Delta g$ is given as a reference for each curve of g_i and g_i'. They are shown in figure 5. The measured quantities are given in table II. The given uncertainties denote the standard deviation of the measured values. Comparing the measured center of gravity with the one we have calculated, we note however, that measured point is

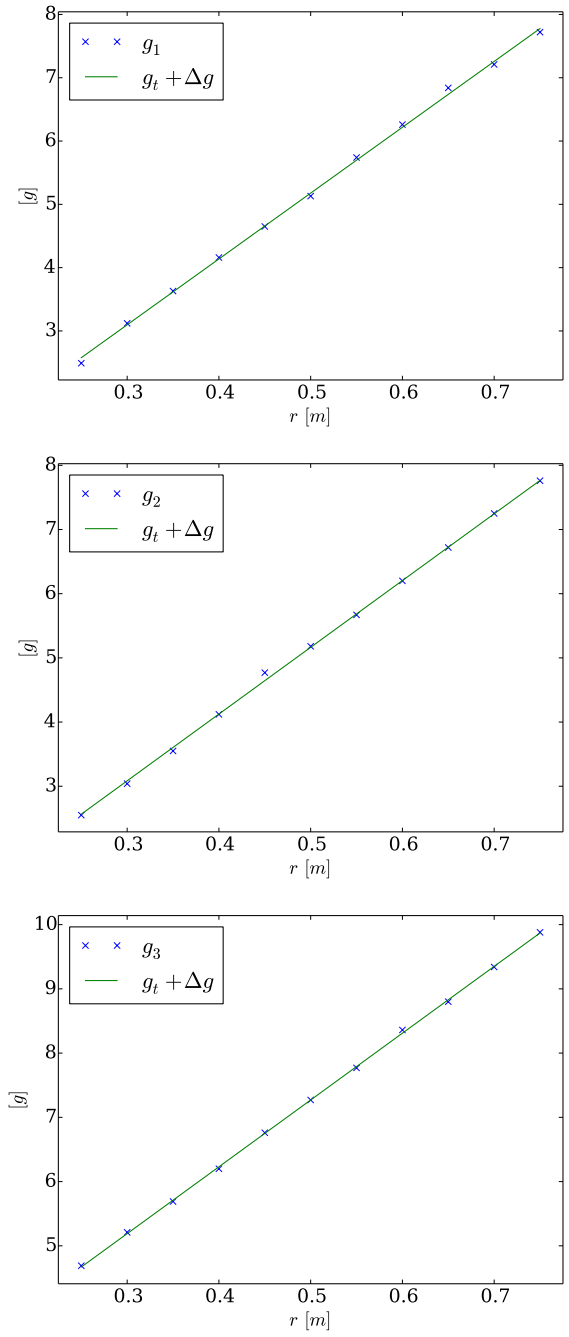

Fig. 4: Measurement curve of the single scale test. The measured curve is shifted by the mean difference Δg from the theoretical prediction g_t, such that the lines are placed upon each other. Only little to no deviation from the prediction can be observed. [`SingleScale.py`]

#	mass [g]	arm	position [cm]
1	10.40	1	25
2	48.07	2	50
3	10.54	3	50

TABLE I: Properties and positions of the objects used for the 2D center of gravity determination test.

	value	rel. σ
$g_2 - g_2'$	21.777(99) g	0.45 %
$g_3 - g_3'$	2.678(69) g	2.56 %
R_x	−119.0(1.6) mm	1.32 %
R_y	−273.3(1.2) mm	0.45 %

TABLE II: Results of the 2D center of gravity determination of 3 test objects. [`CoG.py`]

displaced by roughly 3 mm on each axis, giving a total distance of 5 mm to the predicted center of gravity. This is seen in table III, where $\boldsymbol{R}_m$ is for the measured center of gravity. This can be due to systematic errors in the measurement procedure and has to be investigated further, if more precision is required. A detailed error calculation might be helpful to find out, if this error is due to lack of precision in the determination of the g values.

III. 3D CENTER OF GRAVITY

Since we want to obtain the center of gravity of our unit in three dimensions, we need to combine two measurements of the 2D center of gravity as described in section II-B. For example, by placing the unit in its original orientation, where the z axis points up (here called $z+$), a measurements in the xy plane of the platform gives the x and y components of the 3D center of gravity. See also figure 6. Then rotating the unit by 90 degrees, such that the y axis points downward ($y-$), the same measurement yields the x and z coordinates of the 3D center of gravity Alternatively, we can let the x axis point downward ($x-$) to obtain the z and y component from the measurement.

IV. MOMENT OF INERTIA

The measurement procedure for the moment of inertia is based on the relation [2]

$$I = \frac{R^2}{(2\pi)^2 L} mgT^2 \tag{13}$$

where T denotes the period of the oscillation of the trifilar pendulum. The equation requires, that the center of gravity is

	x [mm]	y [mm]
$\boldsymbol{R}_\mathrm{t}$	−116.6	−269.0
$\boldsymbol{R}_\mathrm{m}$	−119.0	−273.3
$\boldsymbol{R}_\mathrm{t} - \boldsymbol{R}_\mathrm{m}$	2.4	4.3
$\frac{\Delta \boldsymbol{R}}{\boldsymbol{R}}$	2.0 %	1.6 %

TABLE III: Experimental results of the 2D center of gravity determination compared to the theoretical prediction using eq. (2).

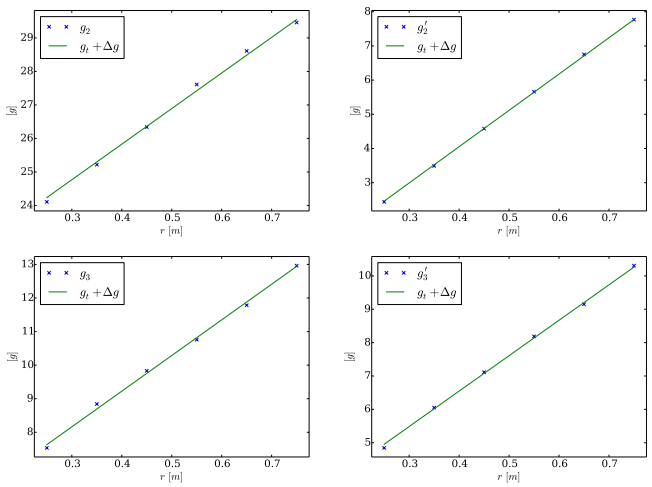

Fig. 5: Measured g curves for the 2D center of gravity test plotted against the prediction g_t for verification of the measurement procedure. [`CoG.py`]

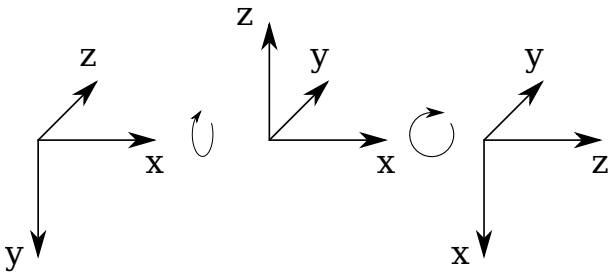

Fig. 6: Possible orientations for center of gravity measurements. Starting with $z+$ in the middle (b), one obtains $y-$ (a) and $x-$ (c) by simple rotations.

in the geometric center of the platform. For the generic case, the factor $\frac{R^2}{(2\pi)^2 L}$ has to be replaced by a more complicated expression. Although, no experimental deviations from this approximated formula have been observed.

To measure the oscillation period, a laser beam points to one arm of the platform. A mirror reflects the beam, which is then projected onto a white paper on the wall and filmed with a camera. A *GoPro Hero* video camera is used to record a video of the laser dot at 240 frames per second. To greatly enhance the contrast of the resulting video file, the room lighting must be turned off during measurement and the desk lamp at the platform used instead. The camera mount and the laser dot is shown in figure 7

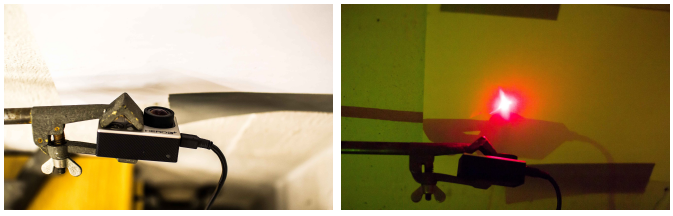

Fig. 7: Mounted *GoPro* camera (a) and projected laser dot with room lighting switched off (b).

Fig. 8: Image from a video capture of the *GoPro* camera. The left side shows the original image and the right side shows the image filtered by the HSV range (150,0,100) – (349,255,255). [`track.cpp`]

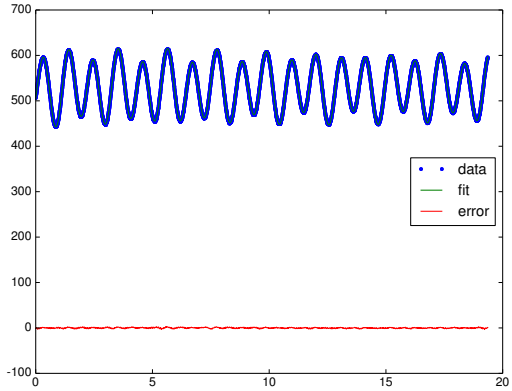

Fig. 9: A sample curve from an tracked video file together with a fit performed with function (14). Also shown is the difference between the fit and the wave function. [`signal.py`]

To analize a video, each frame is converted to *HSV* (*Hue-Saturation-Value*) color space and filtered for the range (150,0,100) – (349,255,255). This selects only the pixels of the laser dot, as demonstrated in figure 8. The first order moments of the filtered image for the x, resp. y axis divided by the total area of the dot then give the coordinates of the point in a particular frame.

The analysis is done by a simple program in *C++*, which uses the *OpenCV* library for image analysis. Since this requires some computing power, analyzing a 20 s video with a resolution of *848 x 480 / 240 fps* takes up to 30 s on an *Intel i5-2540M*.

To determine the period from the analyzed video file, a fit with two damped sine waves is performed. The fitting function is defined as

$$f(\boldsymbol{A},\boldsymbol{\tau},\boldsymbol{T},\boldsymbol{\delta},C,t)=\sum_{i=1}^{2} A_i e^{-\tau_i t}\sin\left(\frac{2\pi}{T_i}t+\delta_i\right)+C \quad (14)$$

Of the two resulting waves, the one with the greater amplitude is taken as the wave to determine T. An example curve, along with the fit is shown in figure 9. The fit is performed as a least-square fit. Its success is heavily dependent on the initial parameters. They have to be adjusted, if the period differs a lot between two measurements.

video duration [s]	I_p [$\mathrm{kg\,m^2}$]	rel. σ
5	0.23479(35)	0.15 %
10	0.23452(29)	0.12 %
20	0.23448(21)	0.09 %

TABLE IV: Computed platform moment and errors derived from the calibration curves. [`Inertia.py`]

A. Characterization

To gather information about the precision of the measurement procedure and to find out, which recording duration of the video measurement is suitable, three series were taken with different recording durations. One series consists of placing two nuts as a test mass with a total weight of 20.94 g on one arm at the positions from 25–75 cm in steps of 5 cm. The chosen recording durations were 5, 10 and 20 s.
Again, we test the measurement against a theoretical prediction. The theoretical moment of inertia of a single object of mass m at distance r is given by

$$I_\mathrm{t} = mr^2 \tag{15}$$

So, as we move the test mass, the measured moment of inertia should rise as r^2.
The moment of inertia of multiple objects is just the sum of the individual moments. Thus, our measurement series is shifted by a constant value compared to the prediction I_t. This constant then simply is the moment of the bare platform, which is finally evaluated in this measurement Thus, by adding the mean difference between the measurement and the prediction to I_t places the two curves on each other, allowing easy verification of the prediction. We denote this quantity with

$$\Delta I \equiv \overline{I - I_\mathrm{t}} \tag{16}$$

The resulting plots of the three measurement series are given in figure 10. As can bee seen in the figures, the measured moments follow the prediction quite well. One can see a slight increase with the longer recording duration. The numerical results for the platform moment are given in table IV, with the uncertainties given as standard deviations of the single measurement series. One can see, that the overall uncertainty of the measurement is in the already range of $\pm 0.00035\,\mathrm{kg\ m^2}$ for a 5 s measurement and reduces only slightly, as the measurement duration gets longer. For comparison, the two nuts, used as test mass, with a total weight of 20.94 g placed at 25 cm produce a moment of inertia of $I \approx 0.0013\ \mathrm{kg\,m^2}$. Therefore, for practical purposes, a measurement duration from 5–10 s can be recommended.

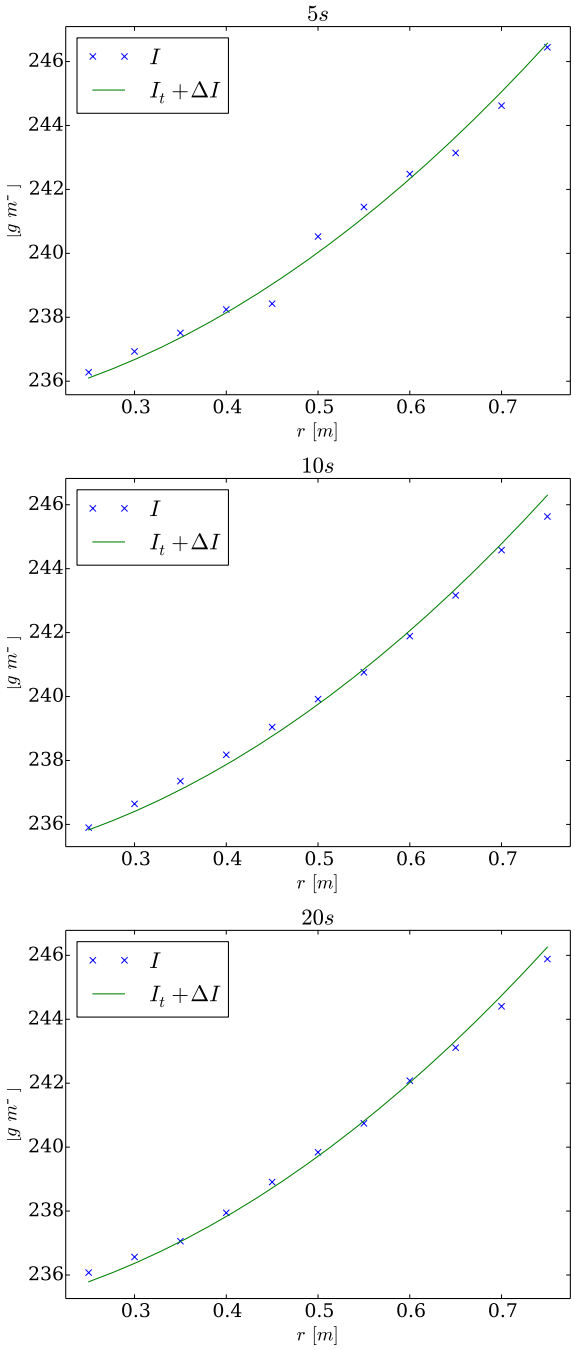

Fig. 10: Measurement series for the inertia characterization with different recording durations plotted against the parabolic prediction I_t of eq. (15). The measurement curve is shifted by the average mean ΔI to place the curves upon each other. [`Inertia.py`]

V. Inertia Tensor

The inertia tensor in matrix representation is given as

$$I = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix} \tag{17}$$

where I_{ij} are the *moment of inertia coefficients*. It thus takes 6 independent measurement to get the full inertia tensor.

Defining an arbitrary normalized vector

$$\boldsymbol{n} = \alpha \boldsymbol{x} + \beta \boldsymbol{y} + \gamma \boldsymbol{z} \tag{18}$$

where $\alpha + \beta + \gamma = 1$. We can write the moment of inertia as [1]

$$I = I_{xx}\alpha^2 + I_{yy}\beta^2 + I_{zz}\gamma^2 + 2I_{xy}\alpha\beta + 2I_{yz}\beta\gamma + 2I_{zx}\gamma\alpha \tag{19}$$

orientation	definition	$\boldsymbol{n}$	I
xx	$\alpha = 1$	$\boldsymbol{x}$	I_{xx}
yy	$\beta = 1$	$\boldsymbol{y}$	I_{yy}
zz	$\gamma = 1$	$\boldsymbol{z}$	I_{zz}
xy	$\alpha = \beta = \frac{1}{\sqrt{2}}$	$\frac{\boldsymbol{x}+\boldsymbol{y}}{\sqrt{2}}$	$I_{xy} + \frac{I_{xx}+I_{yy}}{2}$
xz	$\alpha = \gamma = \frac{1}{\sqrt{2}}$	$\frac{\boldsymbol{x}+\boldsymbol{z}}{\sqrt{2}}$	$I_{xz} + \frac{I_{xx}+I_{zz}}{2}$
yz	$\beta = \gamma = \frac{1}{\sqrt{2}}$	$\frac{\boldsymbol{y}+\boldsymbol{z}}{\sqrt{2}}$	$I_{yz} + \frac{I_{yy}+I_{zz}}{2}$

TABLE V: Orientations with unit vectors and resulting formulas for the moment of inertia measurement.

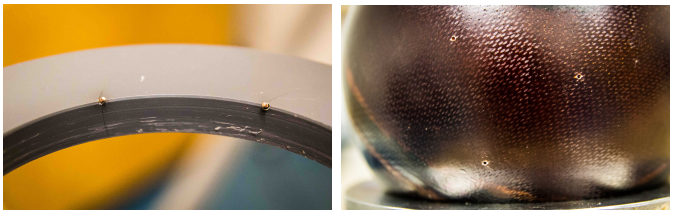

Fig. 11: Small pins attached to the plastic ring (a), that fit into corresponding holes on the unit surface (b) make easy to mount the unit in the 6 different orientations.

Using this representation, we can find a simple set of measurements along an axis to deduct the whole inertia tensor. By measuring the moment of inertia on a diagonal axis, i. e. mounting the unit at an angle of $45°$, we obtain simple expressions for elements. Table V lists the six orientations and resulting expressions for the measured moments of inertia.
From now on, the notation given in the table is used, e. g. orientation xx means measuring along the x axis, whereas xy means measuring along the diagonal axis given by $\boldsymbol{n} = \frac{\boldsymbol{x}+\boldsymbol{y}}{\sqrt{2}}$. Thus, if we want to determine the element I_{yz} and measure I in the yz axis, we first also need to obtain I_{yy} and I_{zz} and then find $I_{yz} = I - \frac{I_{yy}+I_{zz}}{2}$ from our previous measurement.

A. Parallel axis theorem

By simply measuring the moments of inertia in the given orientations, we obtain the inertia tensor in the center frame of the unit. However, the more interesting quantity is the inertia tensor of the unit in the center of mass frame. The parallel axis theorem states that for a body with the center of gravity displaced by the perpendicular distance R, the measured moment if inertia is

$$I = I_c + mR^2 \tag{20}$$

where I_c is the actual moment of inertia in the center of gravity frame of the body. We thus have to subtract the term mR^2 from all our measurements, where R is the distance from 3D center of gravity to the measuring axis. The distance of a point $\boldsymbol{p}$ from a line $\boldsymbol{x} = \boldsymbol{a} + t\boldsymbol{n}$ can be calculated by

$$R = ||(\boldsymbol{a} - \boldsymbol{p}) - ((\boldsymbol{a} - \boldsymbol{p}) \cdot \boldsymbol{n})\boldsymbol{n}|| \tag{21}$$

And since the axis all have their origin in $\boldsymbol{a} = \boldsymbol{0}$, we simply have

$$R = ||\boldsymbol{p} - (\boldsymbol{p} \cdot \boldsymbol{n})\boldsymbol{n}|| \tag{22}$$

VI. Analysis of the Muscat FFU

To apply the whole procedure, the Free-Falling Unit (FFU) from the *Muscat* project has been analyzed. Since the unit has the shape of a sphere, it is easily possible to mount it in the required orientations.
Unfortunately, the FFU did not fit well on the platform, as the three spikes in the middle had a too big distance from each other to support the unit. Additionaly, the spherical shape make it difficult to rotate the unit by exactly 45 or 90 degrees. To overcome this issue, a plastic ring was made, which fits onto the three spike in the center of the platform and has a smaller inner radius, such that the unit fits well onto the ring. In order to mount the unit in the desired angle, two small pins were attached to the ring. On the other side, three tiny holes with a diameter of a about 1 mm were drilled into the sphere. The two pins and the three holes allow us to mount the sphere in six different orientations on the platform, which correspond exactly to the six axes as given in table V. Figure 11 shows the plastic ring on the platform with the pins and the corresponding holes on the spherical FFU. More details on the construction is given in appendix B.
Once the unit can be mounted in the needed orientation, a total of 15 measurement series have to be performed. 8 are required to obtain the 3D center of gravity, 1 for the platform inertia and the additional 6 for the moment of inertia coefficients. This is a very tedious process, and thus could be improved in the future.
As introduced before, we test each measurement series against the prediction, which is

$$g_\mathrm{t} = \frac{mr}{s}$$
$$I_\mathrm{t} = mr^2$$

as given in equations (15) and (11). The mean difference Δg is added, to place the two curves upon each other. A nut of mass 10.55 g was used as a test mass for all series. The results are given in figure 13 for the center of gravity measurements and the inertia tensor measurements. No major deviations from the predicion could be observed. Table VI lists the results from the center of gravity measurements. Note, that the value $g_i - g_i'$ is abbreviated simply as g_i. As one can see, the center of gravity is slightly displaced from the geometrical center of the unit by a about 4 millimeters. Where the relative deviation of the measurements is only about 5 %. However, as seen in the 2D center of gravity measurement in section II-B, there might be some systematic error, so the real center could be shifted by another few millimeters.
The results on the inertia measurements for the single axes are listed in table VII. Also listed there are the values for the parallel axis correction. Comparing these to the actual moments of inertia, their magnitude is very small, since the center of gravity lies close to the geometrical center. The final result of the inertia tensor is given in table VIII. The off-diagonal elements are very small, their values are even below the measured deviation. Thus, we can conclude, that the unit is quite balanced in its geometrical frame.

		rel. σ
	2D center of gracity (x orientation)	
g_1	2.932(50) g	1.69 %
g_3	0.884(22) g	2.54 %
R_x	-1.409(54) mm	3.87 %
R_y	-6.15(10) mm	1.69 %
	2D center of gracity (z orientation)	
g_2	0.730(37) g	5.05 %
g_3	-0.84(10) g	12.46 %
R_x	-2.92(29) mm	9.86 %
R_y	-1.531(77) mm	5.05 %
	3D center of gracity	
R_x	-2.92(29) mm	9.86 %
R_y	2.309(75) mm	3.24 %
R_z	-1.409(54) mm	3.87 %
	Distance from origin	
D	3.73(17) mm	4.67 %

TABLE VI: Complete set of results for determining the 3D center of gravity of the *Muscat FFU*. [`InertiaTensor.py`]

axis	I [g m^2]	rel. σ	R [mm]	mR^2 [g m^2]
xx	2.283(89)	3.91 %	2.70	0.0030
yy	2.189(73)	3.32 %	3.24	0.0043
zz	2.404(97)	4.02 %	3.72	0.0057
xy	2.25(15)	6.80 %	3.96	0.0065
xz	2.384(79)	3.31 %	2.54	0.0027
yz	2.43(20)	8.30 %	3.93	0.0064

TABLE VII: Moment of inertia measurements of the *Muscat FFU* along the axes to compute the inertia tensor. Also given is the perpendicular distance of each axis to the 3D center of gravity, as well as the resulting correction mR^2 from the parallel axis theorem. [`InertiaTensor.py`]

VII. Implementation

The code to analyze the measurement results is written in *Python* and relies heavily on *numpy*, *scipy* and *matplotlib*. Experimental data is read from *JSON* files. The complete project is available in a repository on *GitHub* at [4]. And includes all script, which were used to create the plots in this report. Thus, all the measurements results are also stored in the repository.

The video tracker tool is written in *C++* and thus has to be compiled for the platform in use. The code is mostly adapted from the website [3].

axis	I [g m^2]	rel. σ
I_p	244.9(15)	0.61 %
I_{xx}	2.283(89)	3.91 %
I_{yy}	2.189(73)	3.32 %
I_{zz}	2.404(97)	4.02 %
I_{xy}	0.02(13)	807 %
I_{xz}	0.04(14)	349 %
I_{yz}	0.14(23)	168 %

TABLE VIII: The inertia tensor of the *Muscat FFU*. [`InertiaTensor.py`]

It was tried to keep the code as modular as possible and to reuse existing code. Thus, the actual scripts to perform the calculations in the report mostly consist of reading in the data from the *JSON* files and calling a function to analyze the measurement series. Each section, which contains analysis of measurement results is carried out by another script in the main folder of the repository. The script names, which were used to obtain the values and plots are given for reference on each table and figure shown in this report. Each script outputs the measurement values on the console and writes the diagrams into the `out` folder

VIII. Conclusion

We have developed a method for testing the basic functions of the measurement like measuring the shift of the center of gravity caused by placing an object onto the platform as well as the moment of inertia of the whole system. These have then been combined to measure the center of gravity of the object first in two dimensions, then finally in 3 dimensions. This information helped us to correct the measured moment of inertia coefficients, which where then used to compute the complete inertia tensor.

While there was some deviation between the measured center of gravity in two dimensions and the predicted position, the measurements themselves show only small relative standard deviations in the order of 1 % in the experiments. However, the difference between the experimental value and the prediction has to be investigated more thoroughly in a future work.

The moment of inertia measurement of the platform has been done with a quite high precision, giving only a relative standard deviation of 0.1 %. But since the platform moment is much higher than the that of the actual unit being measured, this precision may not be sufficient for many cases, thus giving a relative standard deviation of up to 10 % on the *Muscat FFU* measurements. This precision however, can propably easily be increased by just capturing longer videos with the *GoPro* camera. Another way to increase the precision might be the use of the general expression of eq. 13.

Finally, all the code is given to combine the measurements to a complete determination of the inertia tensor. Nevertheless, an error analysis of this computation has yet to be done, in order to identify the inputs, that are most crucial in the precision of the single coefficients. This could also be part of some future work.

References

[1] H. Goldstein, C. Pool, J. Safko, *Classical Mechanics*, 3rd Ed. June 25, 2001

[2] Ernest Company Vallet, *A method for determining and balancing the mass properties of the Free Flying Units*, February 11, 2014

[3] http://www.aishack.in/2010/07/tracking-colored-objects-in-opencv/

Source code repository

[4] https://github.com/morloy/trifilar-mass-prop

APPENDIX A
INSTRUCTIONS ON DETERMINING THE INERTIA TENSOR

In this section, we describe the necessary steps to do a full measurement of the inertia tensor.

a) Preparations:

- Choose a test weight, e. g. a small nut.
- Choose the positions to place the nut on a arm for the measurement series, e. g. 30–70 cm in steps of 10 cm.
- Make sure, that you can mount the unit in all required orientations xx, yy, zz, xy, xz, yz, with the center of these axes vertically aligned at the platform center.

b) Center of gravity:

1) Choose two orientations on the main axes, e. g. xx and zz.
2) For each orientation, perform the measurement of the 2D center of gravity:
   a) Place the unit at the platform in the given orientation.
   b) Raise the central spike and the two support points with the scales.
   c) Use a bubble level on each supported arm to align the platform to the xy plane.
   d) For each supported arm, take a measurement series.
      i) For each position, place the test mass and note the value on the scale.
   e) Remove the unit.
   f) Take a measurements series again on each arm, that has been supported when the unit was placed on the platform. The test mass makes the arm stay on the support point.

c) Inertia tensor:

1) Lower the central spike and the support point, such that the platform can move freely again.
2) Point the laser on the mirror on arm 3, such that it is visible on the paper sheet at the wall.
3) Switch off the room lighting and use the desk lamp instead. The laser dot then becomes clearly visible.
4) For each orientation, place the unit on the platform and perform a measurement series.
   a) Place the test mass at the desired position.
   b) Rotate the platform by a small degree, such that the laser dot on the wall moves by about 10 cm.
   c) Record a video with the *GoPro* camera of at least 7 seconds. For this purpose, the WiFi remote is very useful.
5) Remove the unit and perform one measurement series with the platform alone.
6) Copy the video files from the $GoPro$ camera and rename them after the scheme `axis/position.MP4`. The position of the test mass must given in meters with two decimals, e. g. `xx/0.30.MP4`. This means, that there is a seperate folder for each orientation.

d) Use the code: To apply the code on the just taken measurements, the results must be entered in a *JSON* file. As a starting point, the file *FFU.json* should be used, which is also attached to this document for reference. Now follows a short description of the parameters found in the file.

1) Masses are given in *kg* for parameters `unit mass`, `platform mass` and `test mass`.
2) `positions` are given in *m*.
3) Center of gravity measurements in `cog`
   a) `axes` describes the two axes in which the 2D CoGs were measured. Currently supported are "xy", "xz" and "yz". The later measurement series must be given in exactly this order!
   b) `data` holds two measurement series, one for each CoG in the order given in `axes`.
      i) `name` can be freely chosen and is displayed in the output to make the result more readable.
      ii) `free arm` gives the arm, that is not supported by a support point.
      iii) `series` contains the measurement series for g_i and g_i'.
         A) `name` another freely chosable name.
         B) `data` series for g_i in *gram*!
         C) `platform` series for g_i' in *gram*!
4) `initial parameters` have to be given for the least-square fit performed on the filmed inertia wave. These parameters are critical for the success of the fit. If the fit fails, a list of the previously successful parameters is printed. These can be copied into the *JSON* file for the next try. They are given in the order $A_1, \tau_1, T_1, \delta_1, A_2, \tau_2, T_2, \delta_2, C$, as they appear in eq. (14).

When the file is prepared, it has to be stored in the same directory, which holds also the video files from the inertia measurements.

Finally, the script `InertiaTensor.py` can be used. The name and path to the *JSON* file is given in the script itself, and thus has to be adjusted. If the script runs without errors, it should output a list of all variable as they appear in tables VI, VII and VIII.

APPENDIX B
CONSTRUCTING AXES ON A SPHERE

The outer appearance of the *Muscat FFU* consists of two hemispheres and a hole for an electrical connector. The sphere has a radius of $r = 6.2$ mm. Since we need to mount it in 6 different orientations on the platform, there was a need to construct the main axes on the surface, which have a right angle between each other, as well as the diagonal axes, which are inclined at 45 degrees relative to the main axes. Additionaly, these axes have to be aligned to the reference frame of the platform.

The construction was finally carried out using a compass and some geometrical considerations to construct a circle with an arbitrary perpendicular radius on the sphere surface. For this reason, a formula was derived to calculate the radius of compass circle needed to draw a circle of radius R on the surface of the sphere. Looking at figure 12, we want to find the length c in dependence of the radius r of the sphere and the radius R of the circle, we want to construct.

The fomula then can be found as

$$a = \sqrt{r^2 - R^2} \tag{23}$$

$$b = r - a \tag{24}$$

$$c(R) = \sqrt{R^2 + b^2} \tag{25}$$

$$= \sqrt{2r^2 - 2r\sqrt{r^2 - R^2}} \tag{26}$$

$$= r\sqrt{2\left(1 - \sqrt{1 - \left(\frac{R}{r}\right)^2}\right)} \tag{27}$$

Now, by using this formula for $R = r$, we can draw the equator for a given point on the sphere. In this case the equation eq. (27) reduces to

$$c(r) = \sqrt{2}\, r \tag{28}$$

Taking an arbitrary point and drawing the equator and then drawing the next equator from another point on the equator line, we can construct the main axes, which have an angle of 90 degree to each other.

The next step is to draw the diagonal axes. For this purpose, we use $R = r \sin\frac{\pi}{4} = \frac{r}{\sqrt{2}}$ and thus find

$$c\left(\frac{r}{\sqrt{2}}\right) = \sqrt{2 - \sqrt{2}}\, r \tag{29}$$

By drawing circles of this radius around each axis point, we find the diagonal axes at the circle intersections.

Finally, we want to mark the circle, where the plastic disk touches the surface, when the unit is placed on the platform. The plastic disk has an inner radius of $R = 9.14$ mm. To draw this ring on the sphere, we thus need to set the compass to $c \approx 5$ mm.

When drawing these circles around each of the 6 axis points, it has been observed, that they intersect at exaclty 3 points, where each circle has 2 intersection points. This was then the reason to drill tiny holes into the sphere surface and attach pins to the plastic ring. With this setup, the unit can now be mechanically fixed to the platform in one of the 6 orientations.

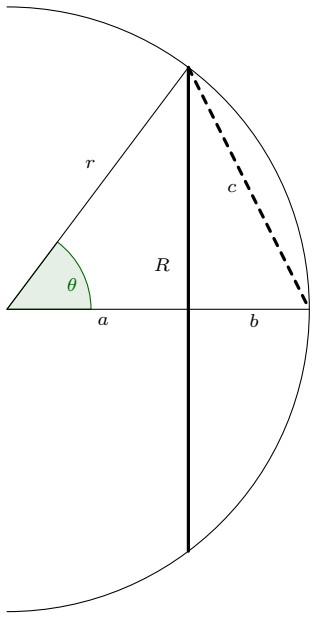

Fig. 12: Cross-section of a sphere with radius r. Here R denotes the perpendicular radius of a circle on the surface and c gives the distance to be used on the compass to construct the given circle.

APPENDIX C
INERTIA TENSOR MEASUREMENTS CURVES

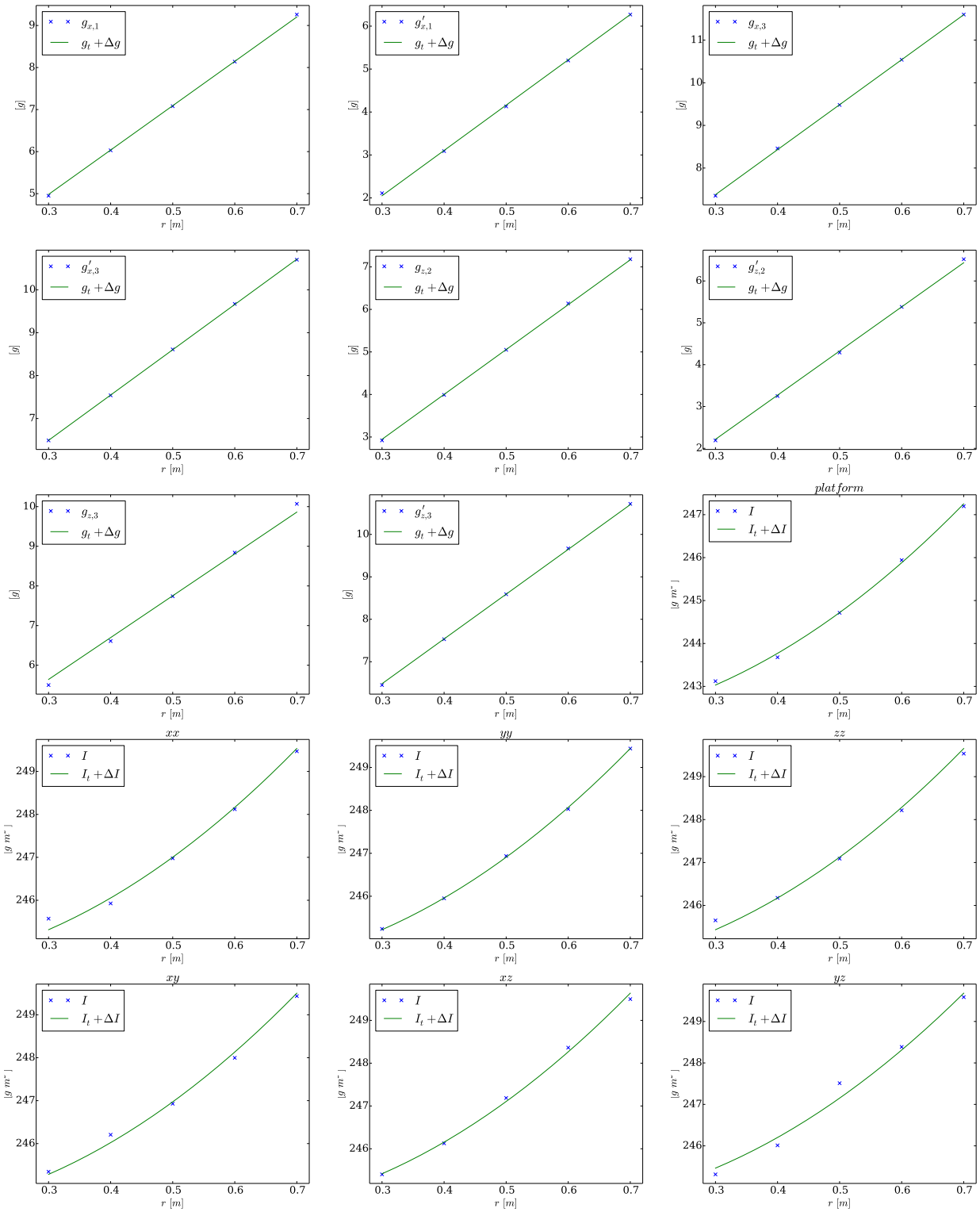

Fig. 13: Complete set of measurement curves for the inertia tensor of the *Muscat FFU*. Shifted upon theoretical predictions g_t and I_t to verify the linear/quadratic shape. [`InertiaTensor.py`]

APPENDIX D
SAMPLE DATA

Listing 1: `FFU.json`

```
{
  "unit mass":  0.413,
  "test mass":  10.55e-3,
  "platform mass":  1.007,
  "initial parameters": [ 7.06236450e+01,  6.60490167e-03,   8.32935942e-01,   1.4422
     9689e+00,   1.48813333e+01,   3.06163121e-03,  -2.00470840e+00,   1.42074061e+01
     ,   4.25097725e+02],
  "positions":  [ 0.3, 0.4, 0.5, 0.6, 0.7 ],
  "cog": {
    "axes": "xz",
    "data": [
      {
        "name": "R_x",
        "free arm": 2,
        "series": [
          {
            "name": "g_{x,1}",
            "data": [ 4.95, 6.03, 7.08, 8.14, 9.26 ],
            "platform": [ 2.11, 3.09, 4.13, 5.20, 6.27 ]
          },
          {
            "name": "g_{x,3}",
            "data": [ 7.35, 8.46, 9.48, 10.54, 11.60 ],
            "platform": [ 6.49, 7.54, 8.61, 9.67, 10.70 ]
          }
        ]
      },
      {
        "name": "R_z",
        "free arm": 1,
        "series": [
          {
            "name": "g_{z,2}",
            "data": [ 2.92, 3.99, 5.05, 6.14, 7.18 ],
            "platform": [ 2.19, 3.25, 4.29, 5.38, 6.52 ]
          },
          {
            "name": "g_{z,3}",
            "data": [ 5.50, 6.61, 7.74, 8.84, 10.07 ],
            "platform": [ 6.45, 7.53, 8.59, 9.67, 10.72 ]
          }
        ]
      }
    ]
  }
}
```