Controlling the position of an acoustically levitated particle

Revised on 25/02/2024

Supervisors:

Junsun Hwang & Lorenzo Noseda (MED 3 2715; junsun.hwang@epfl.ch, lorenzo.noseda@epfl.ch)

Location:

EPFL, Lausanne Campus, MED 2 2419

Keywords:

Acoustic Levitation, Image Processing, Stability Analysis

Requirements:

MATLAB 2021b or higher

1. Introduction

Standing wave acoustic levitation is achieved by generating forces that counteract gravity to trap objects steadily in mid-air. The forces that counteract gravity are called acoustic radiation forces. They emerge due to acoustic radiation pressure, which is the pressure impinging objects in acoustic fields. Due to nonlinear phenomena, the time-averaged components do not cancel out, and apply constant forces that counteract gravity. This result is not trivial, since acoustic waves are harmonic (i.e., sinusoidal), and the time average (Eq.(1)) of a sinusoid is zero, which leads to zero acoustic radiation forces.

$$0 = \langle \sin(2\pi f t) \rangle = \frac{1}{T}\int_0^T \sin(2\pi f t)\,dt, \quad T = \frac{n}{f}, \quad n \in \mathbb{N} \tag{1}$$

However, in our system, the acoustic pressure and frequencies are high enough, such that the acoustic wave is no longer harmonic, and as a result, particles can be trapped in mid-air.

Acoustic levitators are being used as robotic end-effectors to handle delicate objects while avoiding contamination. Acoustic levitators have an inherit limitation when it comes to the speed of manipulation. The levitators create axisymmetric acoustic traps (1). They generate strong trapping force in the direction counteracting the gravity (z), whereas the force is considerably lower in the radial direction (r) (Figure 2). As a result, when the robotic arm translates in any direction other than z, it can only do it relatively slowly without dropping the trapped object. To overcome this limitation, an angular (θ) degree of freedom can be introduced, such that the levitator is tilted while translating (u).

In this laboratory session, you are asked to characterize such a system, and develop an open-loop control strategy based on numerical simulations. The system consists of two acoustic ultrasonic transducers to generate the acoustic traps, two stepper motors to control the translation and rotation of the levitator, a camera for tracking the object and the levitator, a signal generator to excite the ultrasonic transducers, a microcontroller and a workstation.

2. Theoretical background

2.1. Gor'kov potential

Acoustic radiation forces stem from nonlinear phenomena, and estimating the forces in the general case is not straightforward. However, in the case of spherical, rigid particle with a radius (a) much smaller than the acoustic wavelength (λ) there is a simple method to estimate it, using Gor'kov potential. The acoustic wavelength is related to the speed of sound in the medium and the acoustic frequency as follows:

$$c = \lambda f \tag{2}$$

where, c is the speed of sound, and f is the acoustic wave frequency. The 3D Gor'kov potential is computed as follows:

$$U_G = 2\pi a^3\left(\frac{f_1}{3\rho_0 c_0{}^2}\langle p_1^2\rangle - \frac{f_2\rho_0}{2}\langle \mathbf{u}_1\cdot\mathbf{u}_1\rangle\right), \quad f_1 = 1 - \frac{\rho_0 c_0^2}{\rho_p c_p^2}, \quad f_2 = 2\left(\frac{\rho_p - \rho_0}{2\rho_p + \rho_0}\right). \tag{3}$$

Where, ρ_0 and ρ_p are the fluid and the sphere densities respectively, c_0 and c_p are the speed of sound in the fluid and sphere respectively, and p_1 and $\mathbf{u}_1$ are the acoustic pressure and particle velocity at the center of the sphere when the sphere is **not** present (i.e., the acoustic field parameters are computed when there are no particles in the field). For further details, please refer to references[1,2] (see the footnote). The force resulting from Gor'kov potential is computed as follows:

$$\mathbf{F} = -\nabla U_G. \tag{4}$$

To compute the potential, we need to know p_1 and $\mathbf{u}_1$. Ideally, we should measure them, however, it is very challenging, and rarely done in practice. To circumvent the latter, the acoustic field is computed numerically, in the following case the boundary element method (BEM) was used, and especially the MATLAB toolbox OpenBEM[3].

In the setup, the transducers[4] operate at 40 kHz and were positioned in space to create three acoustic traps as shown in 1. The acoustic traps are regions of minimum potential. The resulting nonlinear forces around the trap at the center are shown in Figure 2. A careful examination of 1 and Figure 2 clearly shows that the acoustic trap is not symmetric, and it is much stronger in the z direction than in the r direction.

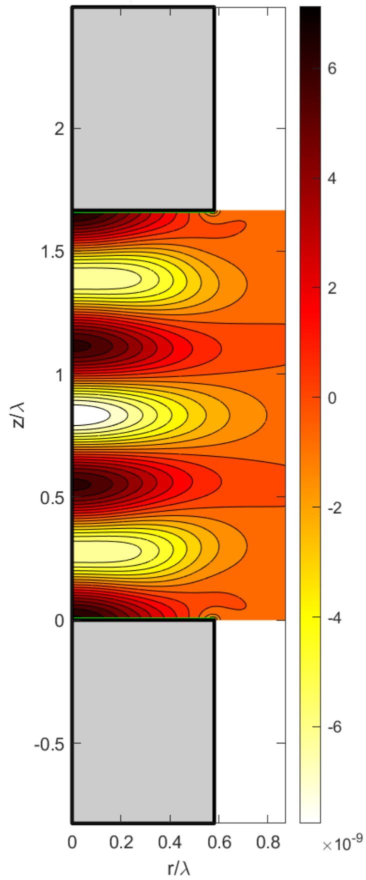


Figure 1. A cross section of the numerically computed nondimensional axisymmetric Gor'kov potential.

[1] Andrade, Marco AB, Nicolás Pérez, and Julio C. Adamowski. "Review of progress in acoustic levitation." Brazilian Journal of Physics 48.2 (2018): 190-213.
[2] Dolev, A., S. Davis, and I. Bucher. "Noncontact dynamic oscillations of acoustically levitated particles by parametric excitation." Physical Review Applied 12.3 (2019): 034031.
[3] http://www.openbem.dk/
[4] https://www.murata.com/en-global/products/sensor/overview/item/ma40s4

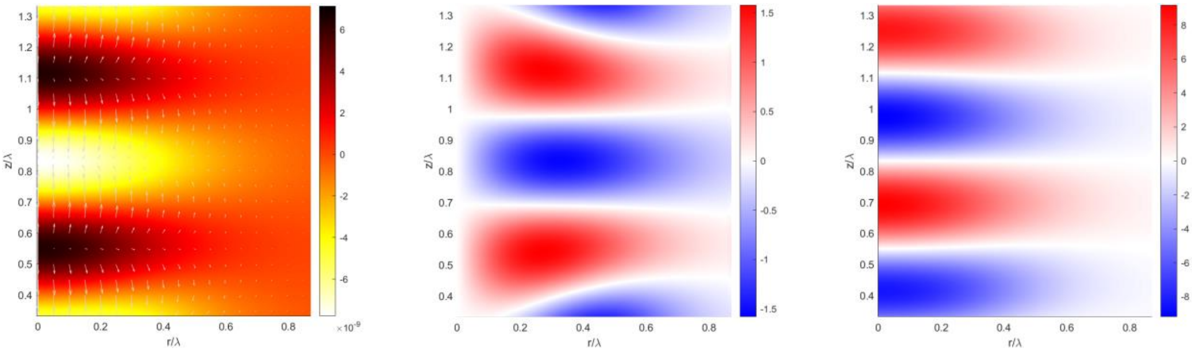


Figure 2. The three panels depict a cross-section of the nondimensional axisymmetric Gor'kov potential U (left), radial force F_r (center) and axial force F_z (right). In the left panel, the arrows indicate the force field.

2.2. Dynamical model – governing equations of motion

The system is modeled as in Figure 3, and we are interested in deriving the governing equations of motion of the levitated particle.

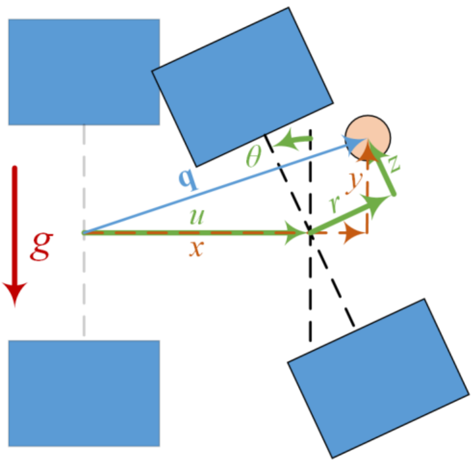


Figure 3. Illustration of the initial and instantaneous configurations of the system.

We assume that the particle is subjected to Gor'kov potential, gravity and drag due to its motion in the air. The drag force is given by the following equation and c_d (drag coefficient) is a function of Re (Reynolds number):

$$\mathbf{F}_d = -\frac{1}{2}\rho_0 \pi a^2 \|\dot{\mathbf{q}}\|^2 c_d(\mathrm{Re}) \frac{\dot{\mathbf{q}}}{\|\dot{\mathbf{q}}\|}, \quad \mathrm{Re} = \frac{\|\dot{\mathbf{q}}\| \rho_0 (2a)}{\mu_0}. \tag{5}$$

Where, $\mathbf{q}$ is a radius vector pointing at the center of the sphere, μ_0 is the dynamic viscosity of the fluid, Re, and the drag coefficient (c_d) can be computed using the following formula[5]:

$$c_d = \frac{24}{\mathrm{Re}} + \frac{2.6\left(\mathrm{Re}/5\right)}{1+\left(\mathrm{Re}/5\right)^{1.52}} + \frac{0.411\left(\mathrm{Re}/2.63\times10^5\right)^{-7.94}}{1+\left(\mathrm{Re}/2.63\times10^5\right)^{-8}} + \frac{0.25\left(\mathrm{Re}/10^6\right)}{1+\left(\mathrm{Re}/10^6\right)}. \tag{6}$$

Using Lagrange equations, the equations of motion can be derived as:

[5] https://pages.mtu.edu/~fmorriso/DataCorrelationForSphereDrag2016.pdf

$$\begin{aligned}\ddot{r} &= -\frac{a^2\pi|\dot{\mathbf{q}}|\rho_0}{2m}c_d\left(\mathrm{Re}\left(|\dot{\mathbf{q}}|\right)\right)\left(\dot{r}+\cos(\theta)\dot{u}-z\dot{\theta}\right)\\ &\quad+\frac{1}{m}\left\{m\left[g\sin(\theta)-\dot{\theta}\left(2\dot{z}+r\dot{\theta}\right)+\ddot{u}\cos(\theta)-z\ddot{\theta}\right]+Vg_r(r,z)\right\},\\ \ddot{z} &= \frac{a^2\pi|\dot{\mathbf{q}}|\rho_0}{2m}c_d\left(\mathrm{Re}\left(|\dot{\mathbf{q}}|\right)\right)\left(\sin(\theta)\dot{u}-\dot{z}-r\dot{\theta}\right)\\ &\quad-\frac{2}{2m}\left\{m\left[g\cos(\theta)+2\dot{r}\dot{\theta}-z\dot{\theta}^2-\sin(\theta)\ddot{u}+r\ddot{\theta}\right]+Vg_z(r,z)\right\}.\end{aligned} \tag{7}$$

Where, Vg_r and Vg_z indicate the partial derivatives of the Gor'kov potential with respect to r and z.

2.3. Control

The objective is to translate the particle as fast as possible (i.e., x coordinate). To do so, we use a linear belt drive (ZLW-0630-Eco) [6], which is motorized by a stepper motor[7] (NEMA 17). In addition, to increase the dynamical stability, we can control the angular degree of freedom, θ, with another stepper motor[8] (NEMA 8). The levitator position and orientation and the particle position in space are estimated by post processing recorded videos.

2.3.1. Sensing – image processing

The sensing is done by processing the recorded videos. We use a digital camera[9] with a fitted lens[10]. To acquire high-quality fast videos (50-100 FPS), the iris and lighting should be adjusted. In addition, a calibration is required to compensate for distortions and obtain the pixel to mm ratio. Each frame of the video is analyzed to track selected features. Here, we use a subpixel image registration algorithm[11] to track two circular markers, and the levitated particle. By obtaining the position of the former, we can estimate the instantaneous position and orientation of the levitator.

2.3.2. Actuation – stepper motors

Each stepper motor is connected to a microstepping driver[12,13] that enables to control the motors easily. We can control the direction (i.e., clock-wise or counter-clock-wise) the size of the step measured in degrees, and when to perform a step. These motors operate in an open loop, which means that it is impossible to know their instantaneous orientation.

2.3.3. Microcontroller

The experimental system is operated and controlled by a microcontroller[14] with an internal clock of 170 MHz. The microcontroller is programmed in C, and using a terminal the user can send text commands that are interpreted for operations such as controlling the rotation velocity of the motors.

In addition, there are limit switches for safety reasons. As soon as one of them is pressed, the motors are turned off, and the levitator moves slowly to the center.

[6] https://drylin-drive-technology-configurator.igus.tools?configurationId=DDX-0001066125
[7] https://www.igus.com/product/1244?artNr=MOT-AN-S-060-005-042-L-A-AAAA
[8] https://joy-it.net/en/products/NEMA08-02
[9] https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca1920-155um/
[10] https://www.baslerweb.com/en/products/lenses/fixed-focal-lenses/basler-lens-c23-1216-2m-s-f12mm/
[11] https://opg.optica.org/ol/fulltext.cfm?uri=ol-33-2-156&id=148843
[12] http://www.motionking.com/download/2L415B_Instruction_Rev.E.pdf
[13] http://www.motionking.com/download/2LD545(FM)_Instruction_Rev.E.pdf
[14] https://www.st.com/en/evaluation-tools/nucleo-g474re.html

2.3.4. Peripheral electronics

The ultrasonic transducers are powered directly by a signal generator[15]. The transducers are fed with a sinusoidal signal at 40 kHz, whose amplitude can be set up to the maximum allowed voltage of 10V.

In addition, there is a 24 V DC power supply which is required for the two microstepping drivers, and a 9 V DC power supply for the LED

2.4. The experimental system layout

An overview of the system is shown in Figure 4. A detailed view of the levitator and the ultrasonic transduces is shown in Figure 5. A detailed view of the linear belt drive and motors is shown in Figure 6, and the microcontroller and microstepping drivers are shown in Figure 7.

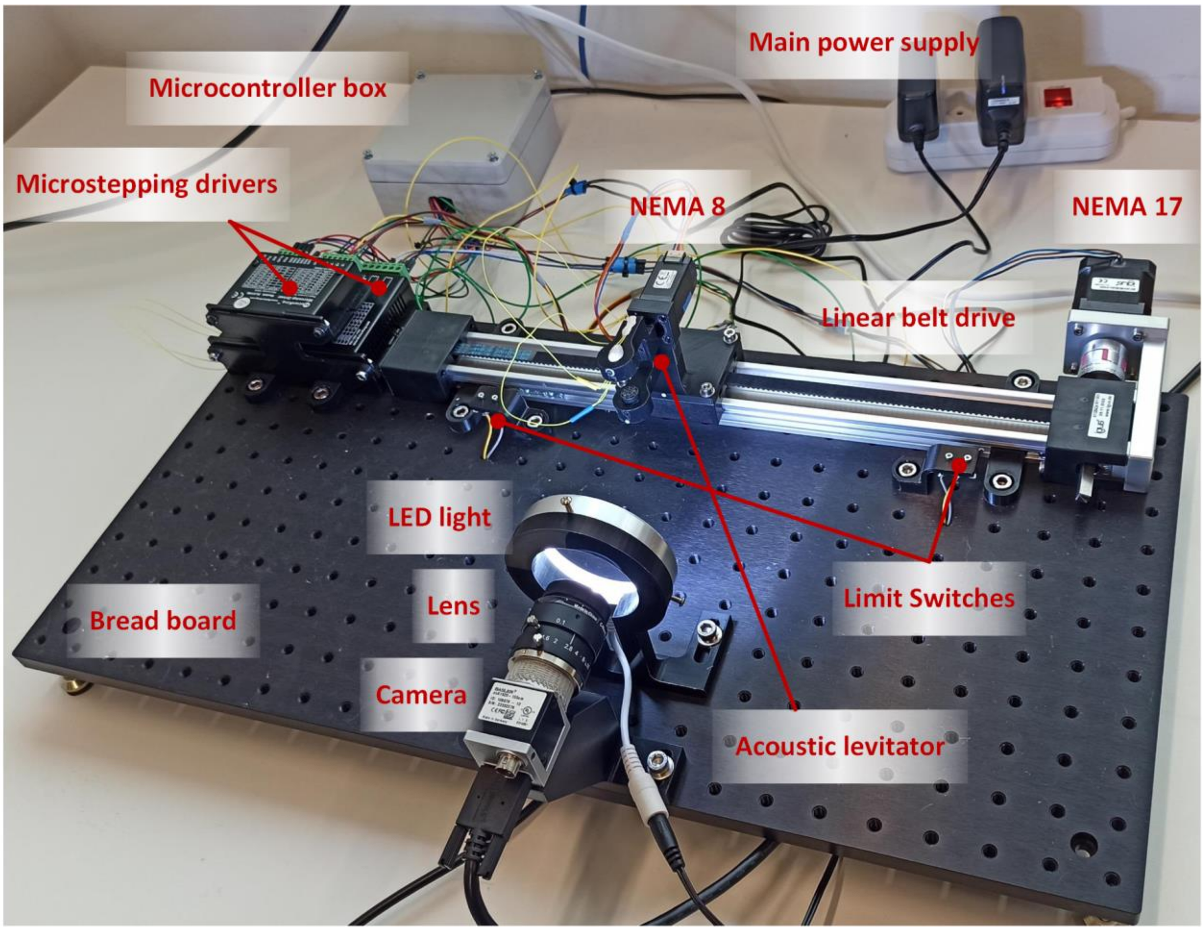


Figure 4. An overview of the experimental system.

[15] https://www.gwinstek.com/en-global/products/detail/AFG-2225

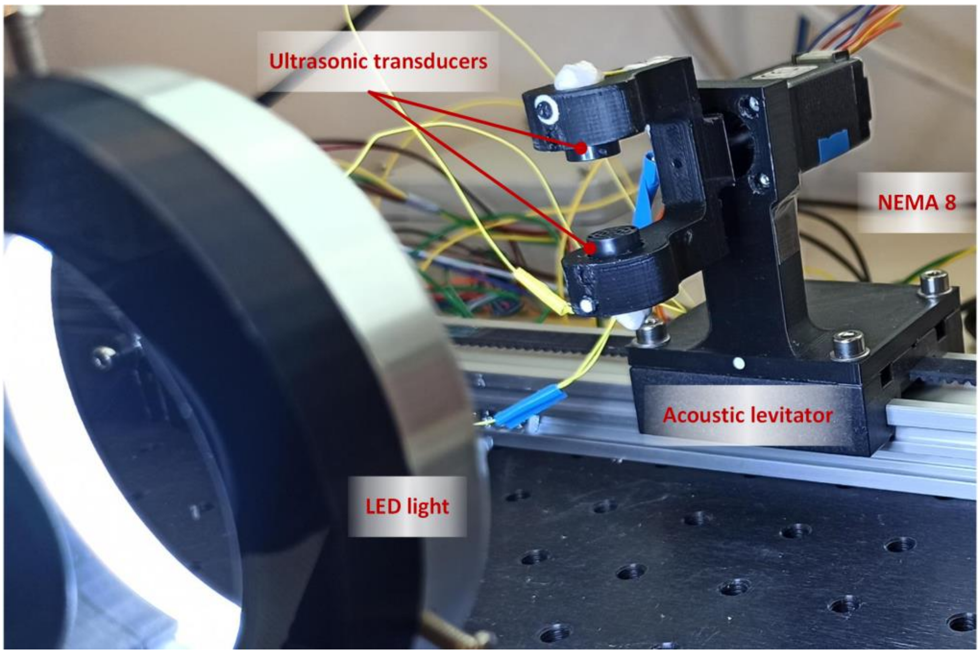


Figure 5. A detailed view of the acoustic levitator and ultrasonic transducers.

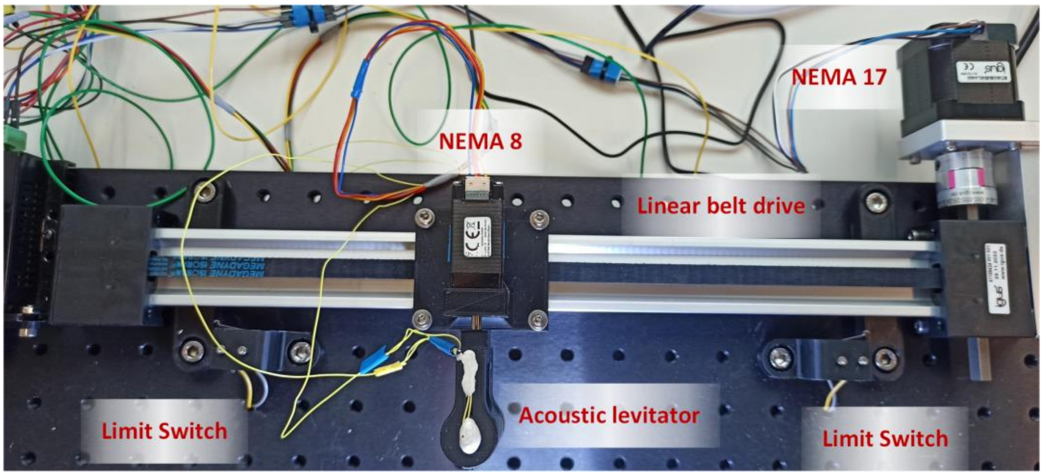


Figure 6. A top view of the linear belt drive, motors, limit switches and acoustic levitator.

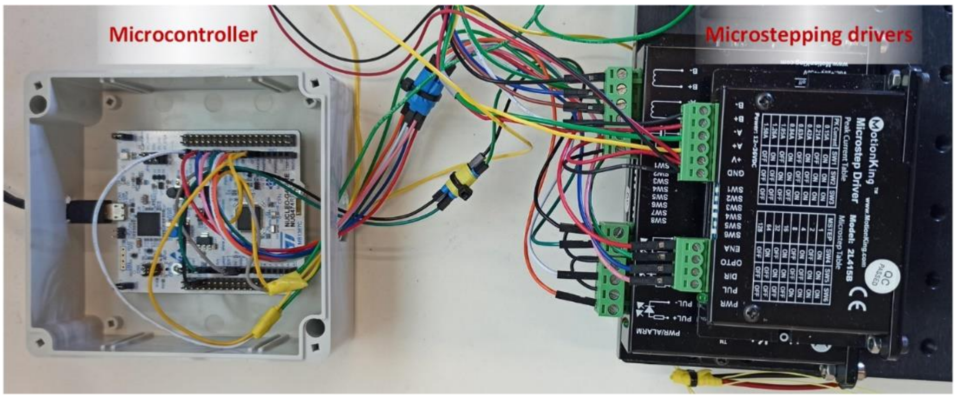


Figure 7. A detailed view of the microcontroller and microstepping drivers.

3. Description of the practical sessions

3.1. Objectives

- Introduction to nonlinear acoustics
- Dynamical modeling of nonlinear systems
- Open loop position control
- Embedded programing
- Image processing

3.2. Preparation

There will be two laboratory sessions. Please read the relevant section carefully before every session, answer the theoretical questions, and test the MATLAB code.

3.3. Evaluation

- The lab is graded according to the evaluation sheet found on Moodle.

The home assignments must be prepared in advance and brought with you to the lab session, you will need them in order to carry out the experiments.

- Answers to the theoretical questions.
- Figures and graphics to support your answers.

The final report should include the followings:

- Title page that includes, full names, SCIPER numbers, date, group number.
- Executive summary (maximum half a page).
- Detailed report including the topics covered in the **home assignments, lab sessions and final report questions**:
    - Provide figures and graphics to support your answer, based on the measured data. Note: do not forget to use legends and units.
    - Provide appropriate explanations along with every figure.
    - State any assumptions and approximations that were made.
    - Provide error analysis where possible.
- Discussion (maximum 1 page).
- Bonus: Suggest improvements to the system and further things to explore.

Remarks regarding the reports:

- The report should be in PDF format, and named following this template:
  tp#_mmdd_spring_2024_group_XX_FR.pdf
  Put the tp number (tp#), and group number (XX).
- The preliminary report **must** be brought to the lab session.

The final report must be submitted via Moodle by 1PM on the Monday following one week after the second lab session.

4. Week 1

4.1. Preliminary homework

Assume that the particle is a sphere with a radius of a = 1.5 mm made of Styrofoam, and the medium is air.

1. Compute the maximum values of the drag force and inertial load when the particle trajectory is given as:

$$x(t)=x_0\sin(2\pi ft),\quad y(t)=0,\quad \mathbf{q}(t)=\big(x(t),\quad y(t)\big). \tag{8}$$

   For x_0 = 15 mm, and a frequency range 0 <f<=5Hz.
   a. Where along the trajectory the maximum drag force value is obtained?
   b. Where along the trajectory the maximum inertial load is obtained?
   c. Show both curves (max forces vs. frequency) separately and on the same graph (i.e., 3 graphs in total).
   d. In the experiment, which one of these forces could lead to the escape of the particle from the trap? Is this the same for each frequency?
2. Compute the values of f_1 and f_2 in Eq.(3).
3. You are not required to derive the equations of motion; however, you are required to derive the following:
   a. The radius vector $\mathbf{q}$, using the coordinates u, r, z and θ.
   b. The kinetic energy, using the coordinates u, r, z and θ. (Ignore the spin of the sphere)
   c. The potential energy, using the coordinates u, r, z and θ. (Do not consider Gor'kov potential).
4. Go to Basler website and use the Lens Selector tool[16]. For the focal lens of the selected lens (see 2.3.1), and a working distance of 100 mm, what is the area in pixel^2 of the sphere's silhouette?
5. Familiarize yourself with the MATLAB Camera Calibrator app[17] (Figure 8 explains how to find it), which is included with the Computer Vision Toolbox (If you don't have it installed you can find it by clicking on the Get More Apps in MATLAB). Read the documentation to understand how to properly acquire images and how to validate the calibration results obtained through the app. You can find 5 images to test out the app on Moodle (the width of each square is 4.583 mm). What can you say about their number, quality and calibration results?
6. As described in 2.3.3, the system is controlled by a microcontroller. Most of its calculations are done with integers and not floating point numbers, as a result only discrete values can be prescribed. For more information, refer to **Error! Reference source not found.**. On Moodle, download 'Valid_fex.fig' and open it in MATLAB as shown in Figure 9, which depicts the frequency values you can prescribe. Choose three valid and three invalid frequencies in the range of]0,5] Hz.

[16] https://www.baslerweb.com/en/sales-support/tools/lens-selector/
[17] https://ch.mathworks.com/help/vision/ug/using-the-single-camera-calibrator-app.html?requestedDomain=en

Figure 8. Camera Calibration app

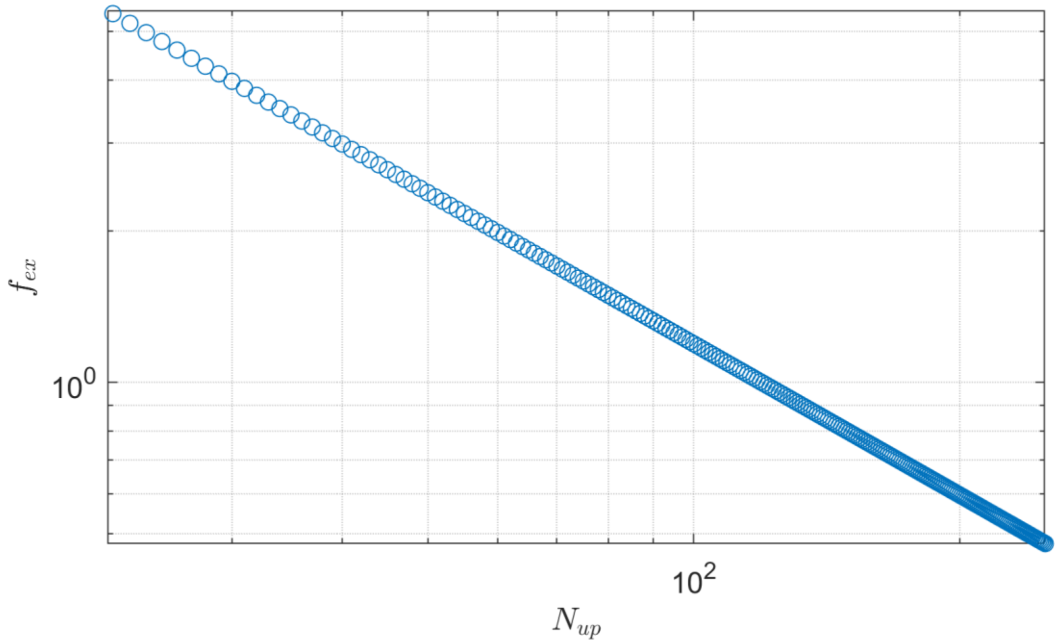


Figure 9. Valid excitation frequency values

4.2. Lab tasks

In this week, you will get to know the system, characterize it and calibrate it.

4.2.1. Image Processing

a. Adjust the camera conditions for iris and focus, and the exposure time to obtain high quality videos at 100 FPS. Once the conditions are found, **lock** the iris and focus.
b. For the chosen camera configurations, record the images which are required for its calibration.
c. Record an additional image (coplanar) located in the same plane as the acoustic levitator to estimate the pixel to mm ratio.

4.2.2. Excitation limitation

Generating an arbitrary trajectory with a microcontroller is not trivial and requires programing a Numerically Controlled Oscillator and a Direct Digital Synthesis**Error! Reference source not found.**. In the upcoming experiments, the trajectories will be either linear (i.e., constant rotating speed) or harmonic, and you will notice that the system cannot oscillate at any frequency and at any amplitude. Here we will explore the limitations.

To control the setup we use the program which is described in Appendix 2. For the following experiments, set the levitator at the center of the field of view, and orient it vertically as shown in Figure 10.

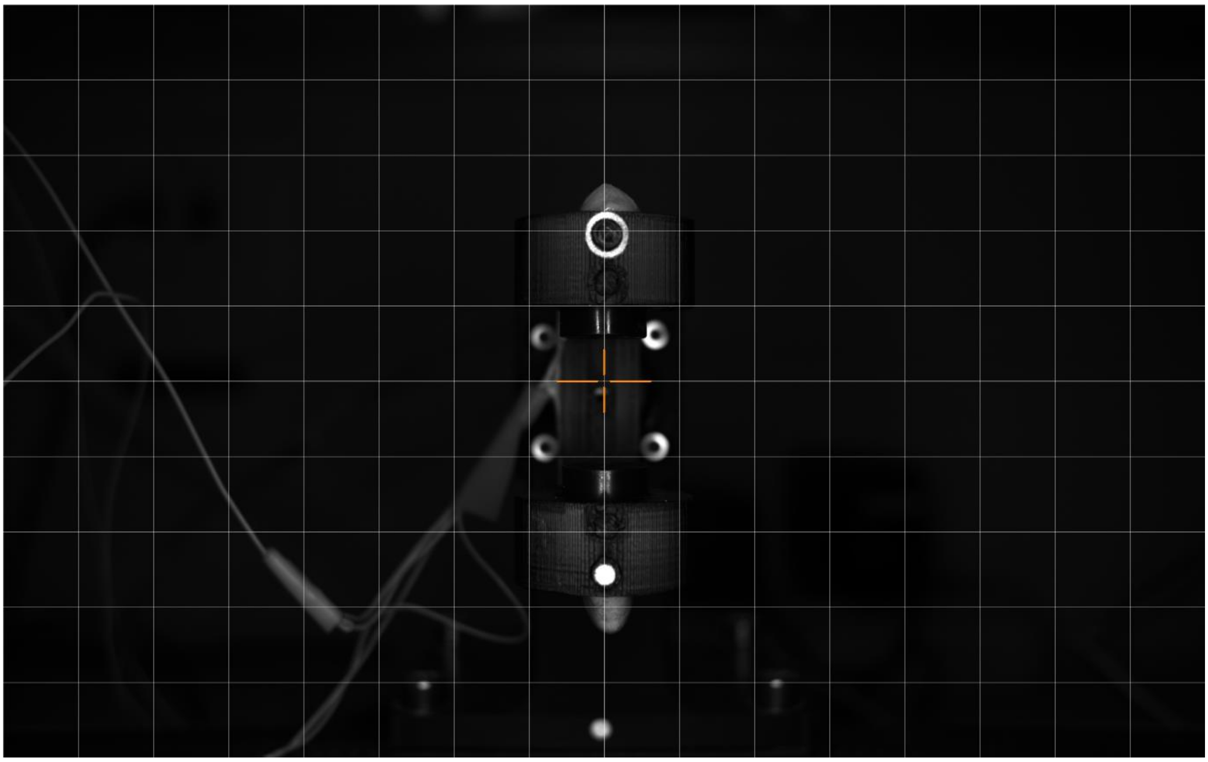


Figure 10. Aligned levitator

a. Using program 2 (`Prog 2`) turn the levitator 800 steps to one side and then 800 steps to the other (`V1:4, S1: ±800`, 10 FPS, video duration 10 s) – record two videos. Extract the transmission rate (i.e., step to turns) in the post processing.
b. Using program 2 (`Prog 2`) move only the linear stage 1000 steps to the left and then 1000 steps to the right (`V2:1, S2: ±1000`, 10 FPS, video duration 5 s) – record the video. Extract the transmission rate (i.e., step to mm) in the post processing.

Turn OFF the power to the motors and light

c. Using program 3 (`Prog 3`) change the frequency of the first motor (`f1`). Set three values that are valid, and three that are not valid according to the graph (Figure 9). For each value, use the command `'Get state'`, to check the actual value and note it down.

Turn ON the power to the motors and light

d. Using program 3 (`Prog 3`), set the translation amplitude to 10 mm (`A2: 10`) and the angular amplitude to 36 degrees (`A1: 36`). Set both frequencies to 2, 3, and 4 Hz (e.g., `f1: 2`, `f2: 2`) and record three videos. (50 FPS, video duration 5 s). Tip: Start recording once the system has reached a stable oscillatory motion.
e. Using program 3 (`Prog 3`) oscillate the system at 1, 3, 4, and 5 Hz, and slowly increase the amplitude until you notice an error. Find the threshold (Amp × f) beyond which the system no longer moves harmonically. Do it separately for each motor. Record the results for both rotation and translation in a separate table format.
   Notice – Maximum allowed values are:
   a. frequency – 5 Hz. (`f1: 5`, and `f2: 5`)
   b. Rotational amplitude – 180 deg (`A1: 180`)
   c. Linear amplitude – 40 mm (`A2: 40`)

4.2.3. Characterizing the acoustic trap(s)

The levitator generates three acoustic traps as shown in 1, the traps are not identical, and apply different restoring forces in every direction.

Choose **three** particles with different sizes (*m1, m2*, *m3*), which you will place at the center trap. Choose two additional particles (*a1*, *a2*) which you will place at the additional sites. **Try to choose the smallest and most spherical particles for the best results.** Avoid deforming the particles by pressing them too hard with the tweezer. Perform the following experiments following the instruction below. The configurations for this experiment are described in the table below.

Table 1. Particle configurations for the first particle, *m1*

	a	b	c	d	e	f	g
Upper trap				*a1*	*a2*	*a2*	*a1*
Middle trap	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*	*m1*
Bottom trap		*a1*	*a2*			*a1*	*a2*
Voltage	8V	8V	8V	8V	8V	8V	8V

Start with *m1*. Set the rotational velocity to 16 (`V1: 16`), the frame rate to 2 FPS, and then record 17s videos. **After completing steps (a-f) with *m1*, proceed to carry out only step (a) for *m2* and *m3*.**

a. Trap particle *m1* in the center. Rotate the levitator slowly 360 degrees (`S1: ±800`) and record a separate video for each direction.
b. Trap particle *m1* in the center, add the first particle (*a1*) out of the two to the bottom trap and repeat a.
c. Repeat b with the other particle (*a2*).
d. Repeat b and c, but now place the particles (*a1* and *a2*) in the top trap.
e. Trap the particle *m1* in the center, add the first (*a1*) and second (*a2*) particles to the top and bottom traps, and repeat a.
f. Repeat e, but exchange the position of the two particles.

4.2.4. Calibration of the numerical model

In the following assignments you will be requested to numerically simulate the dynamics of levitated particles in MATLAB, where the equations of motion (Eq.(7)) are solved. All the parameters can be found with high certainty except for the Gor’kov potential. While its topology (Figure 1) and the resulting forces (Figure 2) can be estimated numerically their value is unknown. To estimate the force, choose three particles with different sizes (*m1*, *m2* and *m3*), which you will place at the center trap.

1) Align the levitator (see Figure 10) and do not move it during the experiment.
2) For each particle, trap it in the center trap and set V = 10 V. Save a **PNG** image.
3) For each particle, reduce the voltage by 0.5 V and save an image until the particle falls off the trap. Do not record the failed condition.

5. Week 2

5.1. Preliminary homework

In the previous laboratory session, you conducted experiments to characterize the system. In light of these results, you are asked to find if it is possible to translate the particles faster by tilting the levitator. If so, how do you suggest to control the translation and rotation of the levitator?

You can control the amplitude, frequency, and relative phase of each degree of freedom (i.e., motor):

$$u = u_0 \sin\left(2\pi f_u + \varphi_u\right), \quad \theta = \theta_0 \sin\left(2\pi f_\theta + \varphi_\theta\right). \tag{9}$$

To develop your control scheme and test it you are provided with a MATLAB code to simulate the system ('Robotics_Practicals_AL_SIM.m'). The code enables you to obtain qualitative results, since the actual acoustic field is unknown and it was numerically computed. As a first step, to obtain a reasonable order of magnitude of the pressure field you need to calibrate the code. Then, the code is ready for you to test different control strategies.

1. Based on the images you recorded in 4.2.1, calibrate the camera.
   a. Run the 'Robotics_Practicals_Camera_Calibration.m' script to generate the calibration data and the pixel to mm ratio, and to save them to a .mat file by exporting camera parameters.
2. Based on your experiments in 4.2.2a and b, extract the motors transmission rate, to do so use the code 'RP_AL_Image_Processing_TR.m' (Appendix 3)
   a. What is the revolution/step ratio (rev/step) you obtained from the image processing?
   b. What is the linear driver transmission ratio (mm/revolutions) you obtained from the image processing, assuming the ratio 800 step/rev is correct?

   Notice the code saves the file Device_Radii.mat (this file is stored in 422a folder) which stores the radial distance of the markers from the center of rotation. This file is required for the additional MATLAB scripts.

3. Based on your experiments in 4.2.4 estimate the prescaler.
   a. Use the image processing code 'RP_AL_Image_Processing_Prs_calib_PNGs' to estimate the orientation on the levitator and the position of the particles.
   b. Use the optimization code 'RP_AL_Estimate_Prescaler_from_PNG.m' to estimate the prescaler. Only use images where the sphere is present and is not too far from the expected position. Open calib_PNG_data.m file.
4. Once the code is calibrated, use the code 'Robotics_Practicals_AL_SIM.m' to simulate different scenarios where $\theta = 0$, and the initial conditions are:

$$u\left(t=0\right) = u_0 \sin\left(2\pi f_u t - \pi/2\right), \quad \dot{u}\left(t=0\right) = 0. \tag{10}$$

   Try to find the stability and instability regions for $0 < u_0 <= 15$mm, $0 < f_u < =5$Hz, for $5 <= V <= 8$V, where the particle radius is similar to what you used in the experiment.

5. Try to come up with a control scheme where you control θ_0, f_θ and φ_θ to make some of the previously unstable regions become stable. Make sure that in your simulation:

$$\dot{u}\left(t=0\right) = \dot{\theta}\left(t=0\right) = 0. \tag{11}$$

Comment – Feel free to edit the MATLAB codes.

5.2. Lab tasks

5.2.1. Stability region: $\theta = 0$

In the first part of the session, you will reproduce the results you obtained in the preliminary report. You are requested to find when the particle is stable, for 0 <u_0<= 15mm, 0 < f_u<= 5Hz, for 5 <=V<= 8V. Record videos, if needed for the final report.

5.2.2. Stability region: $\theta \neq 0$

In the second part you will test the control scheme you developed in the preliminary report. You are requested to find when the particle is stable, for 0 < u_0<= 15 mm, 0 < f_u<= 5 Hz, for 5 <=V<= 8V. Record videos, if needed for the final report.

6. Final report - questions

In addition to reporting the work done in the preliminary homework and in the lab tasks, please address the following questions.

6.1. Week 1

a. Analyze the videos you recorded in 4.2.2d, what are the estimated amplitudes and frequencies you obtained from the image processing? (‘RP_AL_Image_Processing_NLS_LS.m’)
b. What is the threshold beyond which the system no longer moves harmonically for each frequency? Show at least one example per frequency. (‘RP_AL_Image_Processing_NLS_LS.m’)
c. Report the size of the particles you worked with in section 4.2.3. Describe the experiments you did and discuss your observations. (RP_AL_Image_Processing_P.m’)
d. Estimate the acoustic field (4.2.4)
    - Briefly explain the how you estimated the prescaler and its physical meaning.
    - Based on your numerical simulations and experimental results in the second week, comment on the calibration.

6.2. Week 2

a. For 5.2.1 and 5.2.2 separately:
    - Describe your control strategy.
    - Discuss the similarities and discrepancies between the experimental and numerical simulations. (‘RP_AL_Image_Processing_Full.m’).

Appendix 1 – Prescribing an arbitrary path using a step motor and a micro controller

For simplicity we focus on a harmonic function with the possibility to tune its frequency and amplitude.

For a step motor:

$$\dot{\theta} = \theta_0 f_p \tag{A.1.1}$$

Where θ_0 is the stepper motor resolution

$$\theta_0 = \frac{2\pi}{N_{\mu s}} \tag{A.1.2}$$

The frequency f_p in Eq.**Error! Reference source not found.** depends on the interrupt frequency generated by the Numerically-controlled oscillator (NCO).

$$f_p = \frac{\mathrm{FCW}}{2 \times 2^{bit}} f_{\mathrm{OC}} \tag{A.1.3}$$

Where FCW is the Frequency Control Word, and the NCO frequency depends on the clock frequency (170 MHz), timer prescaler (169), and Output Compare frequency divider (99):

$$f_{\mathrm{OC}} = \frac{f_{\mathrm{clock}}}{(\mathrm{PSC}+1)(\mathrm{OC}+1)} \tag{A.1.4}$$

Therefore, if we wish to prescribe the path:

$$\theta = A\sin\left(2\pi f_{ex} t + \varphi\right) \tag{A.1.5}$$

The velocity should be:

$$\dot{\theta} = 2\pi f_{ex} A\cos\left(2\pi f_{ex} t + \varphi\right) \tag{A.1.6}$$

As a result, the frequency control word is estimated as:

$$\mathrm{FCW} = \frac{2 \times 2^{bit} N_{\mu s} f_{ex} A}{f_{\mathrm{OC}}} \cos\left(2\pi k \frac{f_{ex}}{f_{\mathrm{OC}}} + \varphi\right), \quad t \approx k\Delta t = \frac{k}{f_{\mathrm{OC}}} \tag{A.1.7}$$

In order to allow the motor to step and to have sufficient resolution, we discretize the period, thus having 500 values. This leads to the requirement to update the FCW every $\mathrm{N_{up}}$ cycles. To avoid drift, the FCW needs to be updated an integer number of times during one period.

Appendix 2 – Microcontroller program

The communication with the microcontroller is done via a Universal asynchronous receiver-transmitter (UART[18]). To send commands we use a Terminal in the ST integrated development environment – STM32CubeIDE[19]. The communication is limited, and only **precise** commands are interpreted correctly.

Connect the communication with the microcontroller following a brief instruction below, depicted in Figure 11.

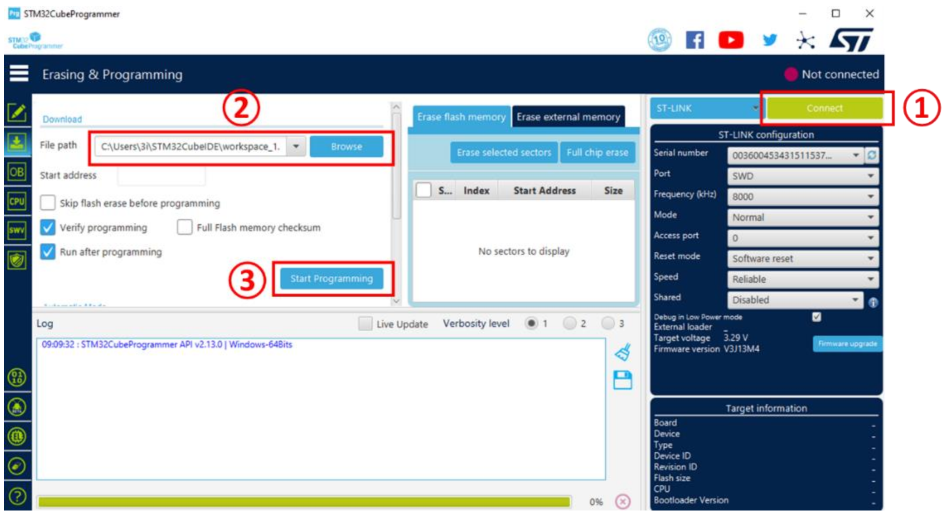


Figure 11. STM32CubeProgramme

1. Open the 'STM32CubeProgramme' software and click 'Connect'.
2. Choose 'C:\Users\3i\STM32CubeIDE\workspace_1.12.0\Robotics_P_V2_NEMA11_NEMA17\Debug\Robotics_P_V2_NEMA11_NEMA17.elf' as the file path
3. Click 'Start Programming'.

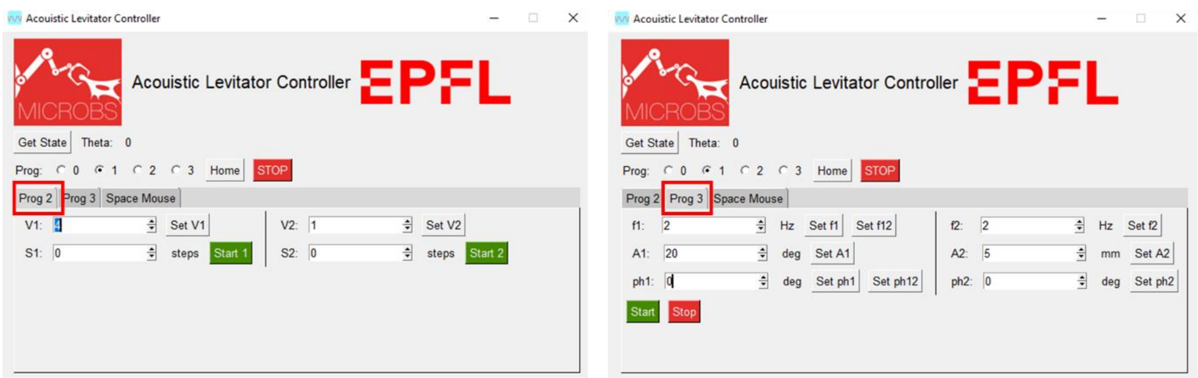


Figure 12. Acoustic Levitator Controller GUI. Prog 2(left) and Prog 3 (right).

Open 'Acoustic Levitator GUI' to control the rotational and translational motions of the levitator, as shown in Figure 12.

Caution: Never input a value of 0 for the frequency!

[18] https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter
[19] https://www.st.com/en/development-tools/stm32cubeide.html

At any given moment the user can probe the system to get the instantaneous values of the parameters by clicking ‘`Get state`’.

Prog 1 – homing

The routine is executed automatically when one of the two micro switches is pressed, and immediately switches to Prog 2.

Prog 2 – manual control

The routine allows the user to control the two stepper motors manually, by modifying their velocity, the number of steppes and the direction.

The velocity is modified by entering values for `V1` and `V2`. The values are frequency dividers, which means that larger input values result in slower velocity.

The number of steps each motor does and the direction is controlled by entering values for `S1` and `S2`. The sign (+ or -) of the value command the rotation direction.

Prog 3 – oscillatory motion

This program enables the user to control each motor harmonic motion separately.

The parameters that can be modified are:

`A1` – Amplitude of motor 1 (units – deg)

`A2` – Amplitude of motor 2 (units – mm)

`f1` – oscillation frequency of motor 1 (units- Hz)

`f2` – oscillation frequency of motor 2 (units- Hz)

`ph1` – phase shift of motor 1

`ph2` – phase shift of motor 2.

Comment – if the ratio between the motor oscillation frequencies is not an integer number, the phase has little meaning.

Appendix 3 – Image processing scripts

Throughout the TP you are asked to use multiple image-processing MATLAB scripts. Some of the scripts only analyze the position and orientation of the levitator and some estimate the position of a levitated particle. To exemplify a correct use of the code, and explain their rationale we demonstrate it through the script 'RP_AL_Image_Processing_P.m'.

Notice all the scripts contain the variable `SAVE`, make sure is it set to `1` to save your results.

Once you run the script you are asked to select the camera calibration results via a window

Select the camera calibration results

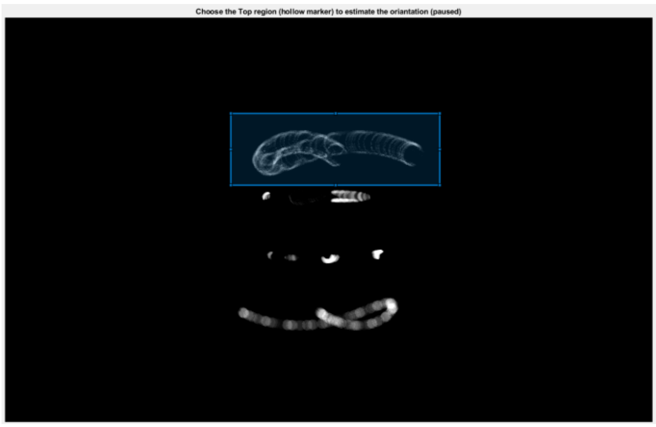
Navigate to where you saved the camera calibration file and click Open. Next, you are asked to load the device radii (this file is generated by the script RP_AL_Image_Processing_TR.m') via a window.

Load the device radii

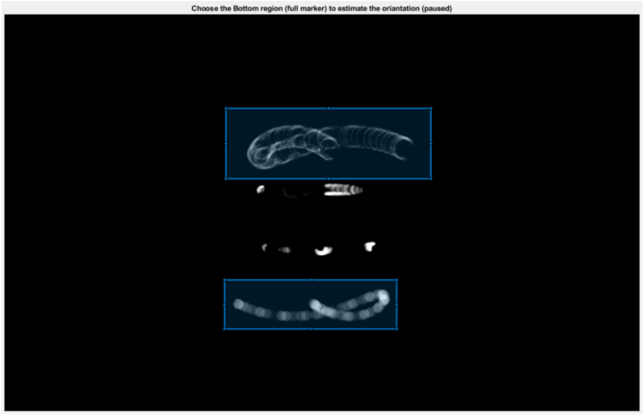
Navigate to where you saved the radii file and click Open. Next, you are asked to load a *.avi file (for 'RP_AL_Image_Processing_Prs_calib_PNGs.m' you are asked to load multiple *.png files).

Two figures will pop, Figure 2 shows the first frame of the video, and Figure 1 shows an average of all the frames. In Figure 1 you are asked to "Choose the Top region (hollow marker) to estimate the orientation (paused)". You need to choose the region where the hollow marker is contained in all the frames, and press Enter.

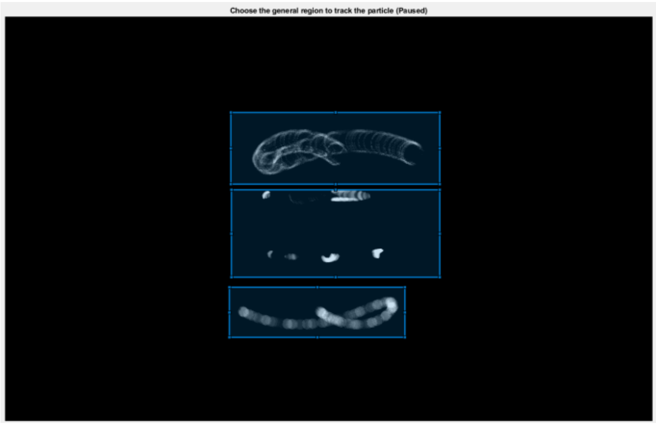
Notice the region can contain the other marker and article.



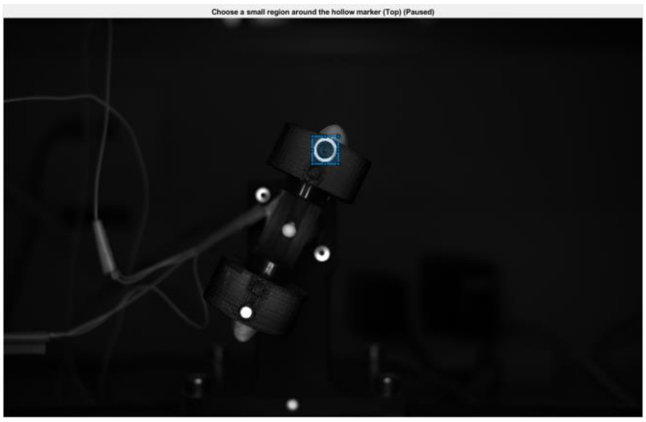
You are then asked to "Choose the Bottom region (full marker) to estimate the orientation (paused)". You need to choose the region where the full marker is contained in all the frames, and press Enter.



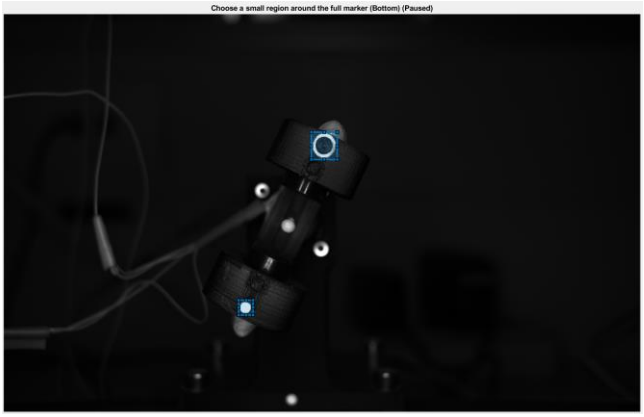
You are then asked to "Choose the general region to track the particle (paused)". You need to choose the region where the particle is contained in all the frames, and press Enter.

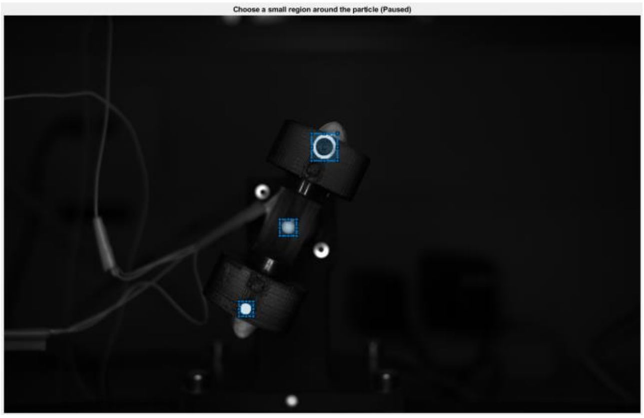


Afterward, Figure 1 depicts the first frame and you are asked to "Choose a small region around the hollow marker (Top) (Paused)". You need to choose the smallest region around the hollow marker is, and press Enter. The pixels in this region are used during the cross-correlation computation.

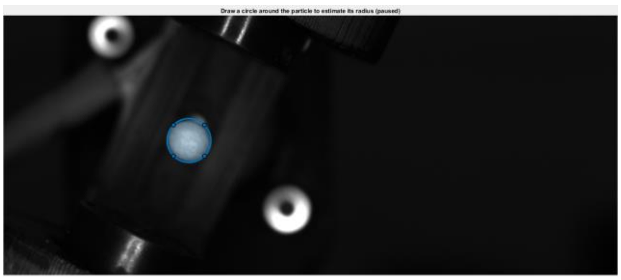


Follow the same procedure for the full marker, and then for the particle.





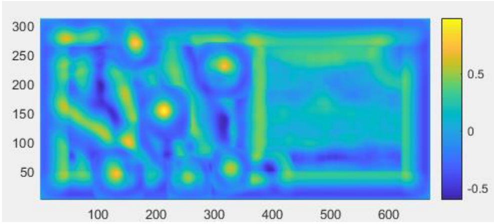
Afterward, Figure 1 depicts the region where the particle will be tracked, and you are asked to "Draw a circle around the particle to estimate its radius (paused)". You are required to draw a tight circle around the particle and press Enter.



Then the script will run and track the position and orientation of the levitator and the position of the particle in each frame. Next, the script analyses the data displays it, and saves it.

Image processing rationale

The implemented image processing implemented in the scripts is simple yet powerful. The algorithm uses cross-correlation (CC) for each frame. It does CC between the original small region of the first frame and the general region in the current frame. The CC operates in X-Y; hence, rotation cannot be found; therefore, circular markers are used. The result from the CC is discrete and limited to the pixel resolution. To increase the resolution and find where the CC value is maximal, a sub-pixel resolution algorithm is used.



First, the maximum value of the CC is found. Second, the 8 pixels around the maximal pixel are found. Third, a paraboloid is fitted to the surface. Fourth, the analytical maximum of the paraboloid is found. Once the analytical maximum is found, its coordinates can be computed with a sub-pixel resolution.