

A New Experimental Technique-Magnetic Particle Tracking and Its Application in Granular Flow

By

Xingtian Tao

Submitted to the graduate degree program in Department of Aerospace Engineering and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chair: Dr. Huixuan Wu

Dr. Brian Kaplinger

Dr. Craig McLaughlin

Dr. Ray Taghavi

Dr. Xuemin Tu

Date Defended: May 13, 2020

The Dissertation Committee for Xingtian Tao
certifies that this is the approved version of the following dissertation:

A New Experimental Technique-Magnetic Particle Tracking and
Its Application in Granular Flow

Chair Dr. Huixuan Wu

Date approved: May 13, 2020

ABSTRACT

Granular flows are omnipresent in nature and they have broad scientific and engineering applications. Current study has benefitted a lot from the development of advanced particle tracking technologies. Some technologies are optical based method (e.g. Lagrangian particle tracking), the accuracy of them is high but they have the limitation in an opaque environment or granular flows; Some techniques use radioactive materials for particle tracking, it can be conducted for granular flow but it is very expensive and require special safe training. We present a new development in the magnetic particle tracking (MPT) technology which can capture the motions of a small particle without optical devices and it is free from radioactive materials. Another advantage of MPT is that it can also acquire rotational information of the particle which other techniques can hardly do. The reconstruction process of MPT using a standard optimization approach is very time consuming, therefore limits its application. In this work, two new MPT reconstruction algorithms are examined and the results are compared with the optimization method. The extended Kalman filter (EKF) algorithm has the same accuracy as the optimization method but is orders of magnitude faster. The speed of sequential importance sampling (SIR) approach is between the optimization method and EKF algorithm. The reconstruct trajectory of this technique is compared with high speed cameras, the accuracy of position obtained using EKF is about 0.6% and the uncertainty of orientation is less than 1.5° .

These advantages of MPT method with EKF algorithm makes it a great tool for studying granular flows. It is applied to a Couette cell to study the dynamics of a tracer particle in a shear flow. Two experiments are conducted with different granular material for various purposes. One is using smaller copper ball with different shear rate the other is a cylindrical plastic tracer with an

aspect ratio of 1 at a constant shear rate. The trajectories of the tracer particle in both experiments show that the grains in a Couette cell with a split bottom moves in a layered structure. The velocity of the particle obtained from the cylindrical tracer is higher in the space between the layers, which suggests that the jumping process between layers is fast. The orientation of particle shows that the cylindrical tracer particle has a preference align in specific directions even though with a low aspect ratio and the shear stress should not cause strong alignment. We use a symbolic based method to examine the jumping and flipping process and the result shows that the durance of particle staying in a preferred state is much longer than in the transition process, and the flipping and jumping are more likely to occur simultaneously. The energy distribution of translation and rotation velocity in discrete zones shows in the near center place these two are approximately the same magnitude. The rotational energy is more stable across different radius while the translational energy varies significantly respect to different radius.

Acknowledgement

I still remember the date, Aug. 11, 2016, when I took a three-day drive from Los Angeles to Lawrence, starting my Ph.D. at the University of Kansas. Time flies and I am about to cross the finish line of this wonderful journey which is one of the most meaningful times of my life. My accomplishments during this close to four-year program would have been impossible without the help of others. I would like to take this chance to express my sincere gratitude to them.

First and foremost, I want to express my deepest gratitude to my advisor Dr. Huixuan Wu. I feel blessed to be his first Ph.D. student and start this wonderful journey with him at Lawrence Kansas. I still remember the first day we met, and he showed me around the department and our labs; help me to settle down and adapt to the new environment. Dr. Wu support me not only by the guidance in my study and research but also my personal life and future development. I appreciate his patient instruction on me, his encouragement for me to explore new things, and his accompany with me physically and mentally when I need it most. His contributions of time and effort enrich my Ph.D. experience and make me this almost four years period happy and fruitful. He is the most important person shaping my career and giving me advice. His incredible passion for research, critical thinking is and will always be a motivation for me.

I also want to appreciate the committee members of my dissertation and professors, staff, officers at the University of Kansas. Thank you for Dr. Brian Kaplinger join me as a new committee member; thank you for Dr. Mclaughlin Craig and Dr. Ray Taghavi's suggestion and guidance upon my comprehensive exam. And thank you for Dr. Xuemin Tu for the cooperation with us on the paper we published and her mentor for me on mathematic skills. Thank you for Dr. Huazhen Fang building the bridge for me with my advisor and the discussion and advice on many

aspects related to my research. Thank you for Wes Ellison's tremendous help on our experiment devices design and manufactures.

I am grateful for the inspiration and encouragement of my friend Longlong Chang and Xiufeng Yang brought to me. The desire to emulate with them is a great part of my pulse to pursuing a Ph.D. degree in this field. I also want to thank all my friends who are caring and supporting me all the time.

I owe my deepest gratitude to my parents and my family, thank you Mr. Tanzhou Tao and Mrs. Xiangzhen Liu. I feel lucky to have parents like you that provide me endless support for my life and study. My aunts, uncles and cousins living in the United States make me feel home in this foreign country. My dissertation is dedicated to everyone I love and those who love and loved me. It is impossible for me to be at this stage without you.

Xingtian Tao

Lawrence, KS April 11, 2020

Table of Contents

ABSTRACT..... iii

Acknowledgement..... v

1 Introduction..... - 1 -

 1.1 Research Objective and Background - 1 -

 1.1.1 Experimental Fluid Dynamics..... - 1 -

 1.1.2 Why Magnetic Particle Tracking - 2 -

 1.1.3 Granular Flows - 6 -

 1.1.4 Couette flow & Couette cell..... - 9 -

 1.2 Organization of Thesis..... - 11 -

2 Feasibility and Preliminary test of MPT - 13 -

 2.1 Analytical Solution of Magnetic dipole Equation..... - 13 -

 2.2 Simulation - 16 -

 2.3 Preliminary Experiment - 18 -

3 Magnetic Particle Tracking Algorithms - 22 -

 3.1 Standard Optimization Method - 22 -

 3.2 Particle Filter - 24 -

 3.3 Extended Kalman Filter - 29 -

 3.4 Comparison of Algorithms - 33 -

4 Experiment and Data Analysis - 39 -

 4.1 A comparison Experiment of MPT Method with High Speed Camera..... - 39 -

 4.2 Error and Accuracy Analysis..... - 48 -

5 Couette Cell Experiment - 53 -

 5.1 Small Diameter Spherical Copper Ball and Spherical Magnet with different shear rate..... - 53 -

 5.2 Cylindrical plastic tracer particle with a constant shear rate - 61 -

 5.3 Small copper ball result..... - 60 -

 5.4 Cylindrical plastic ball result - 66 -

 5.4.1 Position statistics..... - 66 -

 5.4.2 Orientation of the cylindrical particle - 69 -

 5.4.3 A Symbolic Method to Study Flipping and Jumping of the Tracer Particle..... - 71 -

 5.4.4 Relationship between translation and rotation - 76 -

6 Conclusion..... - 81 -

7	Reference	- 83 -
8	Appendix	- 89 -
8.1	Magnetic Particle Tracking Code with Extended Kalman Filter Algorithm (MATLAB)	- 89 -
8.1.1	Read Me	- 89 -
8.1.2	Find the Magnet Moment	- 89 -
8.1.3	Pre-process of The Data	- 91 -
8.1.4	Main Code	- 92 -
8.1.5	SQP Algorithm for Initial Guess	- 93 -
8.1.6	Constrains of SQP algorithm	- 93 -
8.1.7	Extended Kalman Filter Algorithm	- 94 -
8.1.8	Kalman Gain Generator	- 95 -
8.1.9	Velocity Kernel.....	- 96 -
8.1.10	Angular Velocity	- 96 -
8.2	Magnetic Particle Tracking with Particle Filter Algorithm	- 97 -
8.2.1	Particle Filter Main Code.....	- 97 -
8.2.2	Particle Filter Algorithm	- 103 -
8.2.3	Predict State Function.....	- 105 -
8.3	Magnetic Particle Tracking with Sequential Quadratic Programming (SQP) method	- 105 -
8.3.1	Main Code of SQP Algorithm	- 105 -

List of Tables

Table 1	Pros and cons of the widely used particle tracking technologies.....	- 6 -
Table 2	The position and rotation error versus the level of noise	- 38 -
Table 3	Experiment devices and models of the floating cap experiment	- 41 -

List of Figures

Figure 1.	Classification of experiment fluid dynamics method.....	- 1 -
------------------	--	--------------

Figure 2. An illustration of the coordinate system of the MPT. The magnet ball translates and rotates in a certain domain. The cubes stand for an array of magnetometers.- 3 -

Figure 3 some typical granular flows a) granular flow in our daily life b) a fluidized bed is very crucial for many industry such as petroleum, pharmacy etc. c) better understanding the granular motion could prevent the nature disaster such as a land sliding- 7 -

Figure 4 a) 2D Couette cell with coaxial cylinders. b) 3D Couette cell with a split bottom.....- 11 -

Figure 5 A simple setup that has an analytical solution- 13 -

Figure 6 A simulation of a magnetic particle doing an arbitrary translational and rotational motion. The red line stands for the trajectory and the blue arrow represent the orientation of the magnet moment. The boxes stand for the position of the magnetometers.- 17 -

Figure 7 The reconstructed trajectory using MPT method with SQP algorithm in x, y and z directions for a), b) and c)- 18 -

Figure 8 a) and b) are the low-resolution magnet sensor array top view and side view respectively c) is the photo pendulum experiment taking by a high-speed camera d) is comparison of the reconstructed trajectory using SQP with the image based trajectory- 21 -

Figure 9 a state-space diagram- 25 -

Figure 10 a standard particle filter process.....- 27 -

Figure 11 the process of resampling where red dots are eliminated while the upper big green dots become multiple equal weight particle after resampling.....- 28 -

Figure 12. Model underlying the Kalman filter. Squares represent matrices. Ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors. In the simple case, the various matrices are constant with time, and thus the subscripts are dropped, but the Kalman filter allows any of them to change each time step.-

Figure 13. The blue path is the synthetic trajectory of a magnetic ball and the red arrows represent the orientation of the magnet. The square boxes represent the magnetometers in the simulations.

The unit of the axis is meter.- 35 -

Figure 14. Sample trajectory sections reconstructed using the SQP, SIR and EKF with no noise (the left column) and 3% added noise (the right column). a), b) and c) are for x, y, and z component

respectively; d), e) and f) are for M_x , M_y and M_z respectively. The unit of x, y and z is meter and that of M_x , M_y and M_z is Am^2- 37 -

Figure 15. Position error dependence on the noise, and b) The rotation error versus the noise level. -

38 -

Figure 16. a) The calibration of the sensor in a high-speed camera view. This could convert the pixel on the camera CMOS sensors to real distance. Once the calibration and recording the offset of the

test domain finished, the magnet particle is introduced to the test domain. b) A top view of the validation experiment. A cap is floating on the water contained in a plastic bottle. The magnetic ball is glued to the cap center. A black marker near the edge of the cap is used to indicate the orientation of the magnet. Four 3-axis magnetometers are placed near the measurement domain. ...-

41 -

Figure 17. The process of Image Processing to obtain the position and orientation of the magnetic particle a) the original photo at time step k. b) Sum up all the images and doing average to get the

background image of this floating cap c) Post-processing images cropped with only the test field and two tappers are identified with star and square markers- 44 -

Figure 18. The comparison between the reconstructed trajectory and the result from a camera for

a) the x component, b) the y component, and c) rotation angle. d) Trajectory reconstructed by Image processing. e) Trajectory reconstructed using MPT technique with EKF algorithm. The

units in a), b), d), e) are meter, and θ is measured in radian- 47 -

Figure 19 a) The reconstructed coordinates x and y of a fixed ball on the rotating disk with at 10 rpm. b) The magnet moment variation obtained from the MPT. c) The velocity profile of the ball fixed on the rotating disk.....- 52 -

Figure 20 a) The d=3mm copper ball granular flow b) The illustration of Couette cell is discretized into several zones in the radial and vertical directions.....- 54 -

Figure 21. Three sample trajectories. a), b), and c) are corresponding to 8,10 and 20 rpm respectively.- 57 -

Figure 22. Density contours of the tracer particle appearing at each zone for a), b), c), d), e), f), g) are 8,10,12,14,16,18,20 rpm respectively.....- 58 -

Figure 23. Velocity profile showing the ratio of average angular velocity, with different rpm and compared with the error function $\omega(r) = 1/2 + (1/2 \operatorname{erf}((r - R_c)/W))$, here $R_c = 13d$ and $W = 7d$- 59 -

Figure 24 a) A schematic of the experimental setup. b) The diagram of the tracer particle, one coaxial cylindrical magnet is inserted in it. The magnet moment is always aligning with the local z-axis. So, the rotation measured is either around x or y-axis. c) The top view of the surface in the Couette cell. d) Definition of the cylindrical coordinate $(r \theta z)$ and locate frame $(R \Theta Z)$. Here m indicates the magnetic moment.....- 63 -

Figure 25. A sample section of trajectory. The red arrows are the direction of the cylinder particle. The unit of axis is meter.- 65 -

Figure 26. The power spectrum of V_z . Without filtering, the position uncertainty results in high-frequency noise. The Gaussian filter effectively reduces the noise. The frequency unit is Hz.....- 65 -

Figure 27 a) and b) are v_r and v_θ against height at different rpms respectively.....- 61 -

Figure 28. a) The number density distribution along radius direction, this result compressed all data in the z-direction into x-y plane. b) Number density distribution $\rho(r, z)$ obtained from the trajectory of the sample cylinder. The bulk flow shows a layered structure. c) The marginal

probability distribution $\rho(r)$. d) The distribution $\rho(z)$ drawn in the vertical direction. The layers in R direction are labeled as Layer A ($0 < r/R < 0.34$), B ($0.34 \leq r/R < 0.54$), and C ($0.54 \leq r/R < 0.74$). - 68 -

Figure 29 a) The average velocity $\langle V_\theta \rangle$ and b) the average velocity magnitude $\langle |V_r| \rangle$ - 69 -

Figure 30 a) The orientation distribution of the sample particle. The particle prefers to align in the R, Z or Θ direction. The drawing utilizes an open-access Matlab code Ref. b) The contact between a cylinder and particles from another layer helps align the cylinder to be in a preferred direction. - 71 -

-

Figure 31 The status of a particle described using symbols. If a T section is in between two sections with the same symbol, this section is filtered because it is not a real transition (not flipping between different directions). - 72 -

Figure 32. The durance of being in a preferred orientation. It is highly intermittent. b) Histograms of the durance of a preferred orientation or transition process. - 74 -

Figure 33 The probability distribution of all flipping orders in a $S_\alpha \rightarrow S_\beta \rightarrow S_\gamma$ sequence. - 75 -

Figure 34 The histograms of durance of being in a preferred layer and in a jumping process. - 76 -

Figure 35 a), b), c) are Etran, Erot and Total energy distribution contour respectively. d) the energy ratio of Etran and Erot in discrete zones. Error! Bookmark not defined.

List of Publications

Journal Articles:

1. **X Tao, X Tu, H Wu**, “A new development in magnetic particle tracking technology and its application in a sheared dense granular flow”
Review of Scientific Instruments 90 (6), 065116
2. **X Tao, H Wu**, “Quantification of the complexity and unpredictability of a turbulent cylinder wake using excess entropy”
Physica A: Statistical Mechanics and its Applications 523, 211-221
3. Changsheng Wu, Halil Tetik, Jia Cheng, Wenbo Ding, Hengyu Guo, **Xingtian Tao**, Nanjia Zhou, Yunlong Zi, Zhiyi Wu, Huixuan Wu, Dong Lin, Zhong Lin Wang, “Electrohydrodynamic jet printing driven by a triboelectric nanogenerator”
Advanced Functional Materials 29 (22), 1901102

Conference Papers:

1. **X Tao, H Wu**, “A comparison of the sequential quadratic programming algorithm and extended Kalman filter method in the magnetic particle tracking reconstruction”
AIAA Scitech 2019 Forum, 0272
2. **X Tao, H Wu**, “Magnetic-Based Particle Tracking in a Dense Granular Shear Flow”
ASME 2019 International Mechanical Engineering Congress and Exposition
3. **X Tao, H Wu**, “A measurement of particle translation and rotation using magnetic particle tracking”
71st Annual Meeting of the APS Division of Fluid Dynamics

4. **X Tao, H Wu**, “Distinguishing chaotic structures from background turbulence using Benford's law”

APS Division of Fluid Dynamics (Fall) 2017

5. **X Tao, H Wu**, “Modified horizontal visibility graph and its application in turbulence”
AIAA Scitech 2019 Forum, 1868

6. **H Wu, X Tao**, “The evolution of turbulent wake examined using the horizontal visibility graph”
AIAA Scitech 2020 Forum, 0353

1 Introduction

1.1 Research Objective and Background

1.1.1 Experimental Fluid Dynamics

Fluid dynamics has a wide range of applications in our daily life such as determining the mass flow rate of fluid through pipelines, predicting weather, calculating forces on aircraft or submarine. It has several subdisciplines including aerodynamics and hydrodynamics. Experimental fluid dynamics provide us a lot of tools to study it, and through the past decades many experimental techniques have been developed. Here in figure 1 is a summary of the classification of most common experimental fluid dynamics methods. The invasive type: sensor will be in contact with the process flow, while the non-Invasive type: sensor will be mounted externally and there will be no contact with process flow. The intrusive type: sensor will be placed/mounted in the pipeline but either it can be in-contact or no-contact with the process flow. And the non-intrusive type: sensor will be placed/mounted outside the pipeline but either it can be in-contact or no-contact with the process flow.

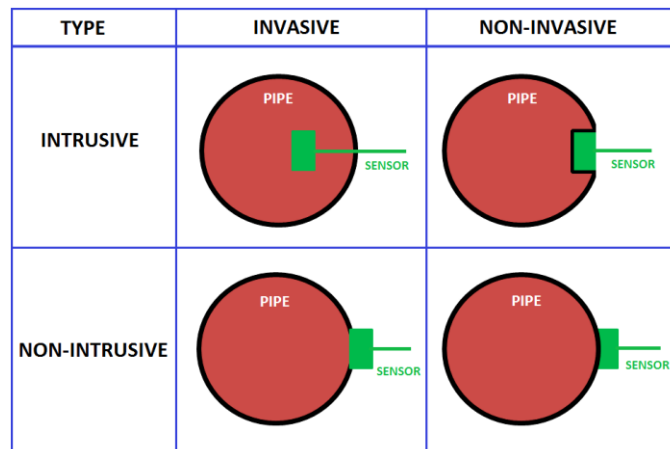


Figure 1. Classification of experiment fluid dynamics method

Each method has its advantage and shortages, ideally, we are seeking a method which will have no influence on the original flow field.

1.1.2 Why Magnetic Particle Tracking

Noninvasive diagnostic method has drawn increasing attention from researchers in the past two decades. New technologies have been proposed and developed. For example, the particle image velocimetry (PIV) can provide 2D or 3D information of entire flow field with high resolution. However, PIV, as an accurate method to measure the translational velocities of particles, can hardly measure their spin. This is mainly due to the seeding particles lacking orientation information. Using magnetic dipole as tracking particles can facilitate the spin measurement because the dipole possesses an orientation. In addition, magnetic particles do not need laser illumination. This magnetic particle tracking (MPT) technique has been applied in some projects in medical field [1]–[3]. The MPT was introduced in the medical field for examining the gastrointestinal tract without the risk of radiation exposure [4],[5]. This is because the human body has the magnetic permeability very close to that of air and exerts very little influence on a static magnetic signal. It is robust and easy to operate with a relatively high accuracy.

The MPT method uses a small magnet as the excitation source. Far away from the magnet, field is modeled as a dipole field given by Equation 1.

$$B(r, r_0, m) = \frac{\mu_0}{4\pi} \frac{[3n \cdot (m \cdot n) - m]}{|r - r_0|^3} \quad (1)$$

where r is the magnet's location, r_0 is the location of a magnet sensor, n is the normal vector in the $r - r_0$ direction, m is the magnetic moment, and μ_0 is the magnetic permeability. Figure 1 is an

illustration of the coordinate and measurement setup. Since a magnetometer usually measures the field component in a particular direction S , the sensor observation should be $B \cdot S$, i.e., the data from the i -th sensor should be

$$D_i = B(r, r_{0i}, m) \cdot S_i \quad (2)$$

where $i = 1, 2, 3, \dots, N$, and N is the total number of sensors. Note that this equation is highly nonlinear. The real observation O from a sensor contains measurement noise and is generally different from this D .

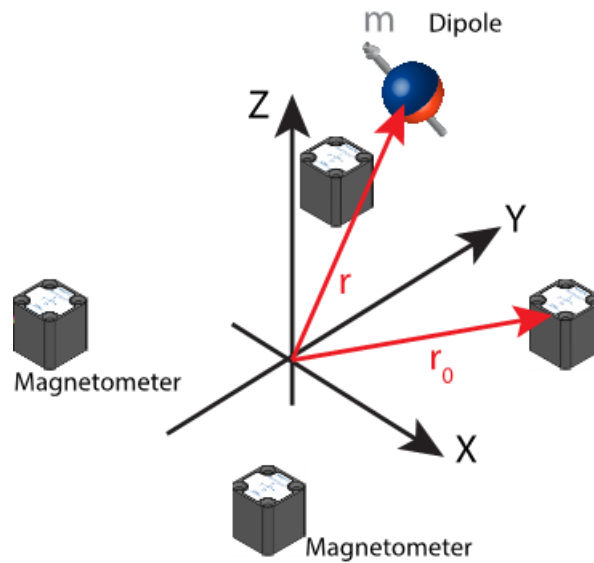


Figure 2. An illustration of the coordinate system of the MPT. The magnet ball translates and rotates in a certain domain. The cubes stand for an array of magnetometers.

Such a magnet's localization is a problem with degrees of freedom 5, and we will discuss more in detail in later chapter 2.1. So, a detection system with at least 5 sensors is required for this problem. Superconducting Quantum Interference Device (SQUID) technique is applied by many researchers to monitor the motion of the magnetic marker. Schlageter[6] and Golden[7] built a system using a 2-D array of Hall sensors for tracking a magnetic particle made of rare earth magnet.

The problem is that the sensitivity of the Hall sensor is too low, and the detected signal becomes very weak as the marker moves away from the sensor array which results in a low signal to noise ratio and low localization accuracy.

It is obvious that the magnet sensor plays a key role for the localization system, the accuracy determines the size of test domain the result of localization. There are many candidate sensors tested by researchers. Hou and Wang tried Hall sensors[8], [9]; Yamada[10] tried Giant Magnetoresist (GMR) sensors; Carr, Daughton and Schotter et al [6], [11], [12] compared the HTS SQUIDs, SDT and GMRs. Back to 2010, Hu et al [5] [3] and Patterson et al [13] began to develop the sensors and made tests in dense gas fluidized beds. Hu used 64 3-axis magnetic sensors to form a cubic sensor array with size $0.5\text{ m} \times 0.5\text{ m} \times 0.5\text{ m}$ and able to capture the motion of a cylindrical magnet as small as $\Phi 4\text{ mm} \times L 5\text{ mm}$.

Besides the applications mentioned in previous, the Magnetic Particle Tracking technique is also applied in chemical industry and many aspects as well. Buist's group[14]–[17] and Erik Sette[18] utilized the magnetic particle tracking technique in fluidized bed.

Because tracking the motion of an object also plays a crucial role in the study of turbulent mixing, granular flow, sedimentation, and cellular biology [19]–[22]. Advanced particle tracking technologies are usually optical based (e.g., Lagrangian particle tracking [23]) and some algorithms are proposed to improve the accuracy[24], [25]. Their precision is high, but they have limited use in opaque multiphase flows or granular motions and it takes a lot of effort to extract the rotational information of the object[26], [27]. A group of non-intrusive and non-optical tracking methods have been developed for opaque environments. For instance, radioactive particle tracking (RPT) and positron emission particle tracking (PEPT) are used to obtain the kinetics of particles in a fluidized bed [28]–[33]. The RPT directly uses the gamma radiation emitted in the beta-decay

of a source particle. The PEPT uses radioactive material such as ^{18}F , ^{61}Cu and ^{66}Ga to label a particle. These materials emit positrons, which annihilate with electrons and generate back-to-back gamma rays. The gamma rays can be monitored by gamma detectors, from which the position of the source can be determined. In addition, X-rays have also been used to measure motion in an opaque environment, such as air bubbles in gas-solid fluidized beds and gas-liquid reactors [34]–[36]. Heindel and coworkers have reviewed the X-Ray applications in the visualization of multiphase flows [37], [38].

However, the methods listed above have shortcomings (Table 1). The optical methods are not operable in opaque flows. Although the RPT, PEPT, and X-rays can be used, they require expensive equipment and special radioactive material operating expertise. In addition, the RPT and PEPT can only monitor the particle's translation, but not the orientation or rotation, which carries important granular dynamics information. The X-ray tomography is only able to detect the orientation of needle-like objects. The limits of measurement technologies and the lack of experimental results have delayed the development of granular dynamics theory and modeling [39].

A magnetic particle tracking (MPT) technology [14], [40] has recently been proposed to address the above issues. In other early efforts, the MPT method was used to track large objects in dense gas fluidized beds with different densities and air flow rates [13], [41], [42]. Recently, Köhler et al. studied the mixing and segregation of fuel in a bubbling fluidized bed, where he used the MPT method to determine the vertical distribution of the tracer in a down-scaled cold unit [43], [44], and Zhang et al. used MPT to measure the particle trajectory in a rotating drum [45]. The far field of a magnet is uniquely determined by the magnet's position and orientation. Therefore, the field strength at multiple locations is sufficient to re-locate the magnet and its orientation. Consequently, the translational and rotational motions of the magnetic particle can be calculated

simultaneously. Compared with other non-invasive techniques, the MPT only needs a few magnetometers and a magnetic bead. Therefore, the cost is significantly lower than the RPT, PEPT and optical-based tracking techniques. In addition, the MPT setup poses no hazardous radiation threat. The advantages and disadvantages of prior techniques are summarized in Table 1.

Table 1 Pros and cons of the widely used particle tracking technologies

Method	Optical-based methods	RPT/PEPT/X-rays	MPT
Cost	Expensive	Very expensive	Cheap
Working condition	Transparent only	Transparent or opaque	Transparent or opaque
Safety requirement	Laser operation	Radiation protection	None
Spatial resolution	High	Relatively low	Relatively low
Rotation motion	Difficult	Generally, no	Yes

1.1.3 Granular Flows

A granular material is a conglomeration of discrete solid, macroscopic particles characterized by a loss of energy whenever the particle interacts. Figure 2 shows some typical granular flow motions, it is related with our daily life, industry applications and prevent of nature disasters.

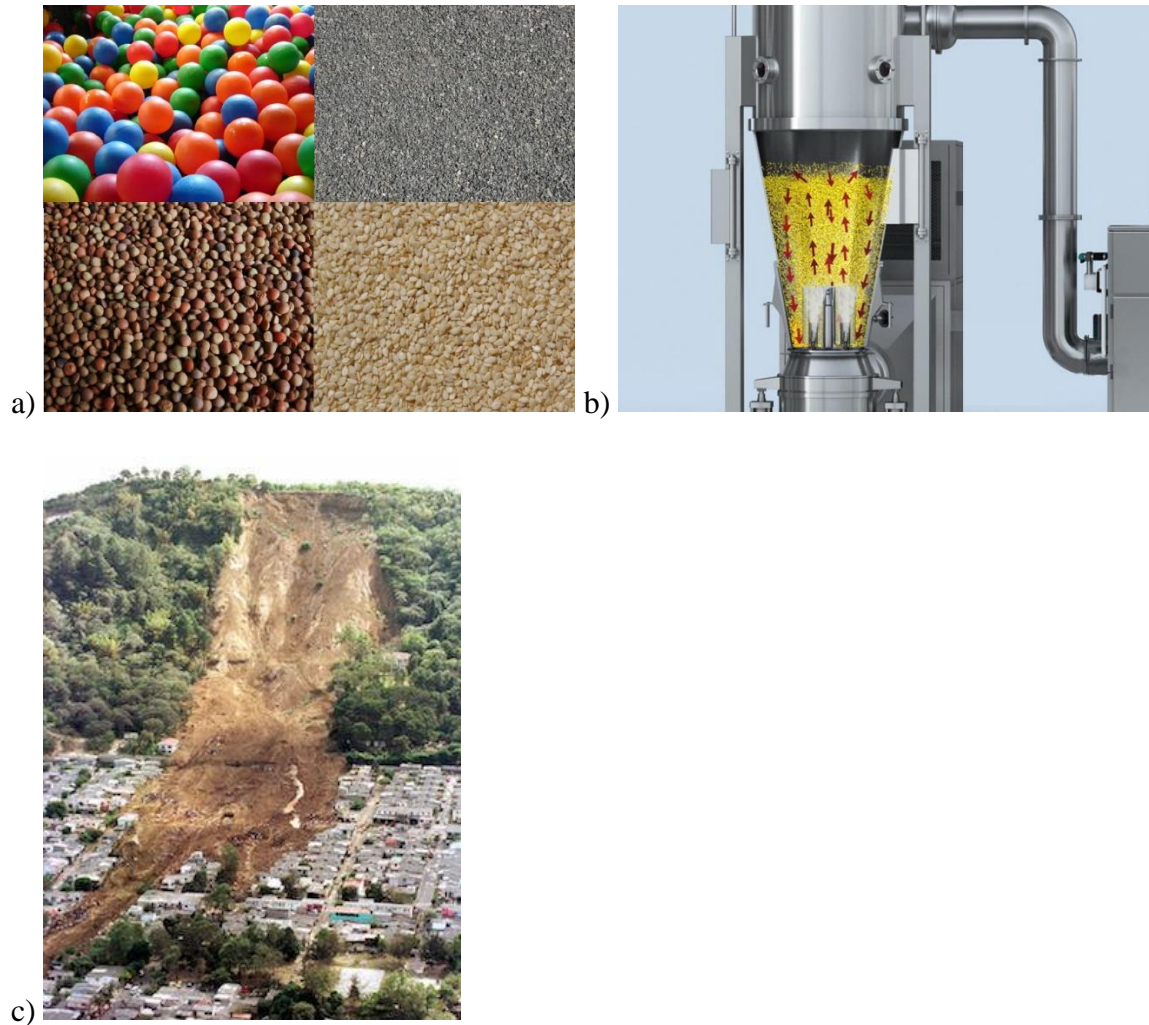


Figure 3 some typical granular flows a) granular flow in our daily life. b) a fluidized bed is very crucial for many industries such as petroleum, pharmacy etc. c) better understanding the granular motion could prevent the nature disaster such as a land sliding.

Granular motions are ubiquitous in our daily life and they have broad scientific and engineering applications. Their dynamics have distinctive properties from homogeneous fluids. A striking feature is their tendency to concentrate shear in narrow bands at low speed.[46] The shear band has unusual properties. For example, the shear zone at the interface between two materials is refracted in analogy with the refraction of a light beam in optics;[47] and surprisingly, the constitutive relation outside the band may depend on the motion in the band[48] (which can be

modeled using global or secondary rheology[49][50]). Understanding dense granular flow remains a major challenge. Since the pioneering work of Bagnold,[51][52] researchers have gradually established models for spherical particles.[53]–[56] These models have successfully captured many aspects of granular flows. However, the particle shape, poly-dispersity and other microscopic parameters are critical to the macroscopic flow behavior.[57] It is not clear how these microscopic properties and dynamics influence behavior on a large scale, and a comprehensive model is needed to describe macroscopic flow based on the dynamics of a single particle.

Much effort has been devoted to studying the motion of individual particles. In the low speed regime, the particle density is high and the durable contact usually dominates the momentum transfer.[58] Many experiments and simulations are performed in split-bottom geometries, e.g., the split-bottom Couette cell,[59][60] where the shear band is near the split. The translational motion of a particle in both surface and bulk flows is accessible experimentally, thanks to the development of high-speed videoing and magnetic resonant imaging (MRI).[57], [59], [61], [62] Perfect monodisperse spheres may crystallize in the shear zone.[63] As the filling height increases, the core that rotates with the bottom disk transits from a trumpet shape to a closed bubble near the bottom.[60] The velocity profile can be fitted using the error function.[64] The shear rate and velocity fluctuation obey a power-law relation $\delta v \propto \dot{\gamma}^\alpha$. [61]

Recently, the dynamics of non-spherical particles has drawn a lot of attention, since the particle shape affects the macroscopic dynamics and most of the particles in nature and industry are non-spherical.[63] Both experimental and simulation results show that the elongated particles align with the streamline direction in the shear zone.[65], [66] The orientation distribution is related to the aspect ratio, and independent of the shear rate.[67] Particles with an aspect ratio of 1 do not have a significant alignment. The orientation order of elongated particles is similar to the director

dynamics in nematic liquid crystal, which can be described using Leslie-Ericksen equations;[65] however, the underlying physics is drastically different, as in the liquid crystal case, thermal fluctuation plays a key role.

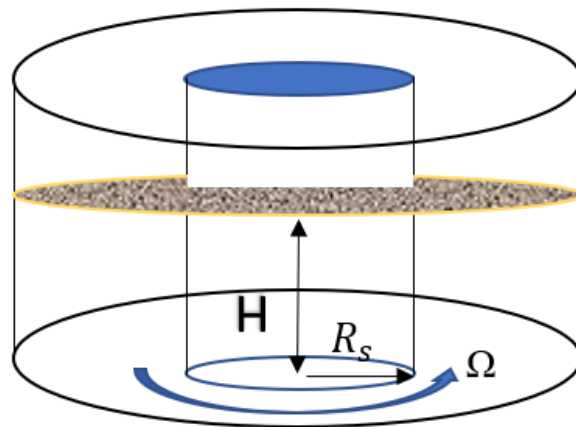
In fact, the motion and orientation of the non-spherical particles are largely determined by the lasting contact among the neighboring particles. In this research, we will show that a cylinder with an aspect ratio of 1 can be aligned in the preferred directions. The alignments are not directly caused by shear. In addition, our results provide the relationship between rotation and translation, along the Lagrangian trajectory of a particle. Such a trajectory over the entire experimental time in a Couette cell has rarely been reported before. These measurements are based on a novel technology, the magnetic particle tracking (MPT).[44], [45] The measurement principle is to reconstruct the motion of a magnetic particle within the field that varies with time. Therefore, both translation and orientation information are acquired simultaneously. Since a magnetic field can easily penetrate non-magnetic materials, the measurement can be used in an opaque environment like granular flows. The magnetic signal is harmless and the setup is not complicated. Therefore, the MPT technology is much safer than X-ray tomography or gamma-radiation based methods.[38] Recently, the authors have improved the reconstruction algorithm using extended Kalman filter and considerably increased the processing speed.[40], [68]

1.1.4 Couette flow & Couette cell

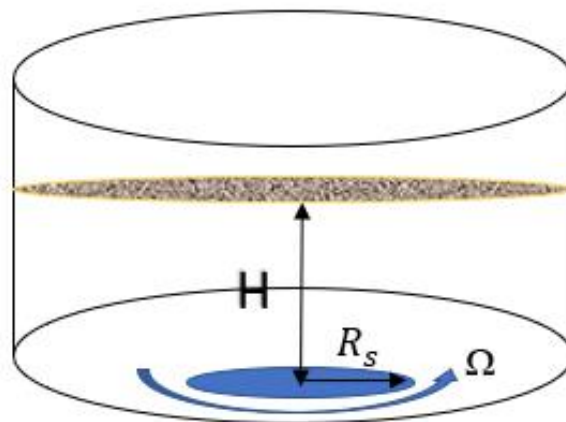
In fluid dynamics, Couette flow is the flow of a viscous fluid in the space between two surfaces, one of which is moving tangentially relative to the other. The configuration often takes the form of two parallel plates or the gap between two concentric cylinders. The flow is driven by virtue of viscous drag force acting on the fluid but may additionally be motivated by an applied pressure

gradient in the flow direction. This type of flow is named in honor of Maurice Couette, a professor of Physics at the French University of Angers in the late 19th Century.

Couette flow is often used in physics or engineering to illustrate shear-driven fluid motion. The Couette cell is a device that can provide constant shear motion continuously. Here in the Figure 3 shows 2D and 3D Couette cells. The height of granular material vs the radius of rotating disk H/R_s is called filling height. Usually the critical filling height is 0.55.



a)



b)

Figure 4 a) 2D Couette cell with coaxial cylinders. b) 3D Couette cell with a split bottom.

1.2 Organization of Thesis

This chapter gives some background of experimental fluid dynamics method and granular flows. And have a brief introduction of why magnetic particle tracking technique is a powerful tool in studying granular flows. Chapter 2 talks about the principle of magnetic particle tracking and magnetic dipole equation. One simulation and one preliminary experiment we did are presented to demonstrate the feasibility of MPT technique.

Chapter 3 is mainly discussing about algorithms to solve the magnetic dipole equations. A standard optimization method, one Monte Carlo method – Particle filter and extended Kalman filter are investigated in each chapter 3.1, 3.2 and 3.3. Chapter 3.4 compares the robustness against different noise level, data processing speed and accuracy of the three algorithms.

Chapter 4 focus on the experiment part of the Magnetic particle tracking method. One experiment is conduct use both high speed camera and MPT technique to compare the result. This experiment is a quasi-2D with both translational and rotational motion. And the accuracy and error analysis are studied.

Chapter 5 discusses Couette cell experiment with different type of granular material for varies purposes and some interesting phenomenon we found in this topic. Chapter 5.1 talks about the small diameter spherical ball in the Couette cell with different shear rate. Chapter 5.2 talks about one cylindrical plastic particle serves as the tracer in the Couette cell with a constant shear rate. And chapter 5.3 and 5.4 discuss the result of the two experiments. Statistic method and a symbolic

based method is used to study the orientation of the cylindrical particle which is also included in chapter 5.4.

2 Feasibility and Preliminary test of MPT

2.1 Analytical Solution of Magnetic dipole Equation

The MPT method uses a small magnetic dipole to generate the magnetic field and a series of magnetic field sensors are installed to record the magnetic field. The sensor readout can be used to reconstruct the position and orientation of the dipole. The magnetic field of a dipole is

$$\mathbf{B}(\mathbf{x}_0) = \frac{\mu_0}{4\pi} \frac{[3\mathbf{n} \cdot (\mathbf{m} \cdot \mathbf{n}) - \mathbf{m}]}{|\mathbf{x} - \mathbf{x}_0|^3}, \quad (3)$$

where \mathbf{x} is the magnet's location, \mathbf{n} is the normal vector in the $\mathbf{x} - \mathbf{x}_0$ direction, and \mathbf{m} is the magnetic moment, μ_0 is the magnetic permeability. The dipole field is exact for a uniform spherical magnetic bead [69], and it is a very good approximation for the far field of non-uniform or non-spherical beads. The far field means $|\mathbf{x} - \mathbf{x}_0| \gg d$, and d is the bead diameter.

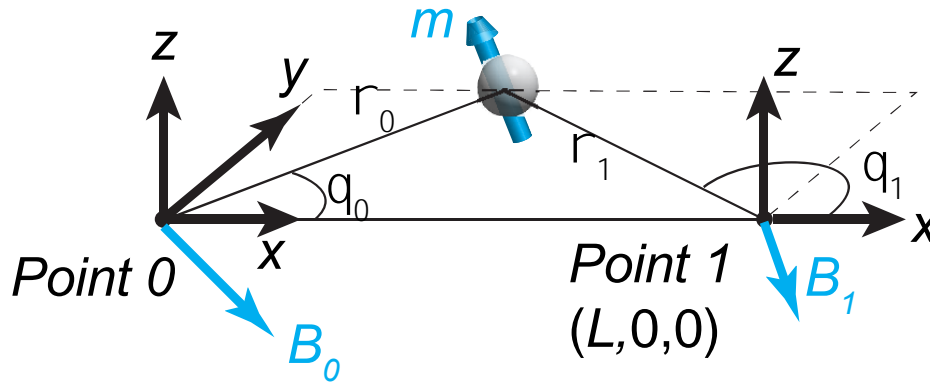


Figure 5 A simple setup that has an analytical solution

In a simple setup, we can analytically reconstruct the particle's position and orientation based on the measured magnetic strength. In this setup (Figure 5), there are two 3-axis magnetometers, one located at Point 0, $\mathbf{x}_0 = (0, 0, 0)$ and the other at Point 1, $\mathbf{x}_1 = (L, 0, 0)$. They together provide

6 signals (B_{0x}, B_{0y}, B_{0z}) and (B_{1x}, B_{1y}, B_{1z}). The frame of reference is constructed such that the magnetic particle is in the x - y plane. Therefore, for Point 0, $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$ lies on the x - y plane and $\mathbf{n}_0 = (n_{0x}, n_{0y}, 0)$.

$$\begin{pmatrix} B_{0x} \\ B_{0y} \\ B_{0z} \end{pmatrix} = \frac{\mu_0}{4\pi|\mathbf{x}-\mathbf{x}_0|^3} \begin{pmatrix} 3n_{0x}(m_x n_{0x} + m_y n_{0y}) - m_x \\ 3n_{0y}(m_x n_{0x} + m_y n_{0y}) - m_y \\ -m_z \end{pmatrix} \quad (4)$$

Consequently, we use the z -component to normalize the other two equations:

$$\mathbf{T}_0 = (B_{0x}/B_{0z}, B_{0y}/B_{0z})' = -3\rho(\mathbf{M} \cdot \rho) + \mathbf{M} \quad (5)$$

where $\mathbf{T}_0 = (T_{0x}, T_{0y})$ can be calculated from the measurement results. Variables $\rho = (n_{0x}, n_{0y})' = (\cos \theta_0, \sin \theta_0)'$ and $\mathbf{M} = (m_x/m_z, m_y/m_z)'$ are the unknowns. The prime here means transpose, and the dot means the inner product of two vectors. Decomposing \mathbf{T}_0 along the ρ direction, $\rho \cdot \mathbf{T} = -3\rho \cdot \rho (\mathbf{M} \cdot \rho) + \mathbf{M} \cdot \rho = -2\mathbf{M} \cdot \rho$, as $\rho \cdot \rho = 1$, and along the transverse direction, $\rho \perp \cdot \mathbf{T} = \mathbf{M} \cdot \rho \perp$, where $\rho \perp = (-n_{0y}, n_{0x})$ and $\rho \cdot \rho \perp = 0$. Reorganizing the terms, we obtain $\rho \cdot (\mathbf{T} + 2\mathbf{M}) = 0$, and $\rho \perp \cdot (\mathbf{T} - \mathbf{M}) = 0$. Therefore,

$$\cos \theta_0 (T_{0x} + 2M_x) = -\sin \theta_0 (T_{0y} + 2M_y)$$

$$\sin \theta_0 (T_{0x} - M_x) = \cos \theta_0 (T_{0y} - M_y)$$

Hence, $(T_{0x} + 2M_x)(T_{0x} - M_x) = -(T_{0y} + 2M_y)(T_{0y} - M_y)$. The same relation applies to Point 1, i.e. $(T_{1x} + 2M_x)(T_{1x} - M_x) = -(T_{1y} + 2M_y)(T_{1y} - M_y)$. These quadratic equations describe two overlapped circles. The solutions are

$$M_x = \frac{m_x}{m_z} = M_x^* \pm \frac{K(T_{0y}-T_{1y})}{\sqrt{(T_{0x}-T_{1x})^2+(T_{0y}-T_{1y})^2}} \quad (6)$$

$$M_y = \frac{m_y}{m_z} = M_y^* \mp \frac{K(T_{0x}-T_{1x})}{\sqrt{(T_{0x}-T_{1x})^2+(T_{0y}-T_{1y})^2}} \quad (7)$$

where $M_x^* = \frac{(T_{0x}-T_{1x})P-(T_{0y}-T_{1y})E}{(T_{0x}-T_{1x})^2+(T_{0y}-T_{1y})^2}$, $M_y^* = \frac{(T_{0y}-T_{1y})P+(T_{0x}-T_{1x})E}{(T_{0x}-T_{1x})^2+(T_{0y}-T_{1y})^2}$, and $K = \frac{1}{2}[T_{0x}^2 + T_{1x}^2 + T_{0y}^2 + T_{1y}^2 + (T_{0x} + T_{1x})M_x^* - 4M_x^{*2} + (T_{0y} + T_{1y})M_y^* - 4M_y^{*2}]^{1/2}$. Here we define two new variables, $P = T_{1y}^2 - T_{0y}^2 + T_{1x}^2 - T_{0x}^2$, and $E = (T_{0x}T_{1y} - T_{1x}T_{0y})/4$. Note that, there are two solutions, one of which is not physical.

Once \mathbf{M} is known, we can deduce the angle $\theta_0 = \tan^{-1}(T_{0y} - M_y)/(T_{0x} - M_x)$ and $\theta_1 = \tan^{-1}(T_{1y} - M_y)/(T_{1x} - M_x)$. These angles, together with the distance L , determine the magnet's position

$$x_p = L \tan \theta_1 / (\tan \theta_1 - \tan \theta_0)$$

$$y_p = \tan \theta_0 \cdot x_p$$

Thereafter, the equation $B_{0z} = -\mu_0 m_z / 4\pi |\mathbf{x}_p - \mathbf{x}_0|^3$ (or $B_{1z} = -\mu_0 m_z / 4\pi |\mathbf{x}_p - \mathbf{x}_1|^3$) provides m_z , and $(m_x, m_y) = m_z \cdot \mathbf{M}'$. The other solution cannot satisfy both B_{0z} and B_{1z} equations, and should be discarded.

In a nutshell, the MPT is to solve a set of equations $\mathbf{B}(\mathbf{m}, \mathbf{x}_p) = \mathbf{D}$ (which is the measured signal) to find the dipole moment and position. The above analytical solution only exists in simple magnetometer arrangement. For general unsymmetrical configurations, it is hard to find an analytical expression. Hence, numerical optimization is needed. In reality, a magnetic sensor usually measures one component of \mathbf{B} , i.e., $S_i = \alpha \mathbf{B} \mathbf{e}_i$, where α is the coefficient for sensor signal to magnetic moment, and \mathbf{e}_i stands for the projective direction of the i -th sensor. The goal is to minimize the cost function $F = \|\mathbf{S} - \mathbf{D}\|$. Here $\|\cdot\|$ means L^2 norm. Several optimization method have been tested [5], [13], [14], [70], [71]. And Sequential Quadratic Programming (SQP) stands out to be the most feasible methods among the optimization method. SQP is a method for

constrained nonlinear optimization. It solves a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. If the problem is unconstrained, then SQP reduces to Newton's method for finding a point where the gradient of the objective vanishes.

2.2 Simulation

One simulation is run to test the reconstructed trajectory using MPT method with an optimization method to solve the magnetic dipole equation. A magnetic particle randomly translates and rotates in the test domain as shown in figure 5. The red line represents the trajectory of this magnetic particle and the blue arrow represent the orientation of the magnet and those square boxes represent magnetometers. In real situation the measurement from magnetometers containing noises, so we intentionally adding two levels of gaussian noise (3% and 10% of the magnitude of the measurement) to test performance of this algorithm.

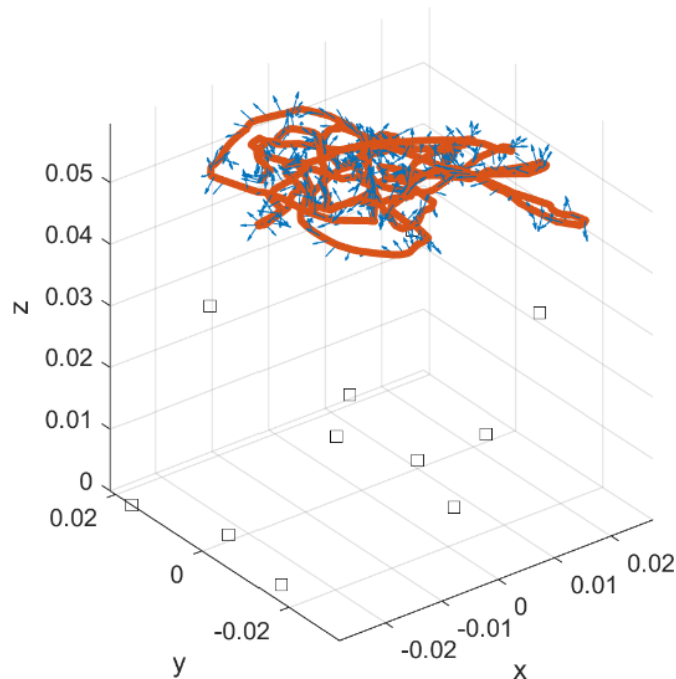
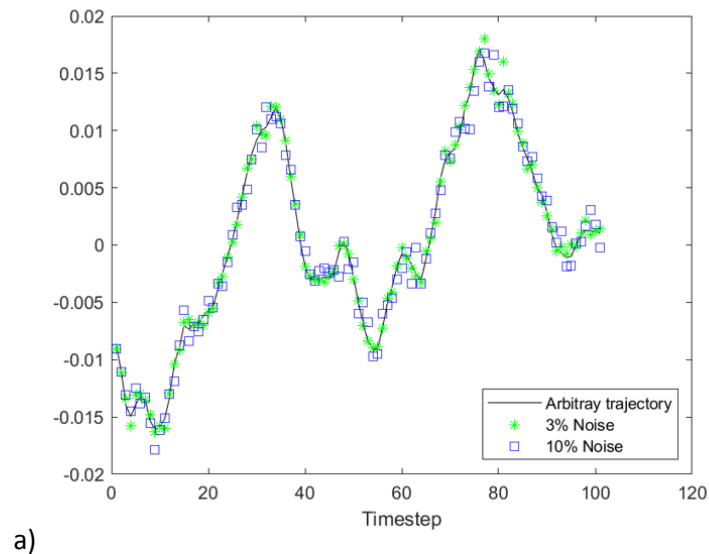
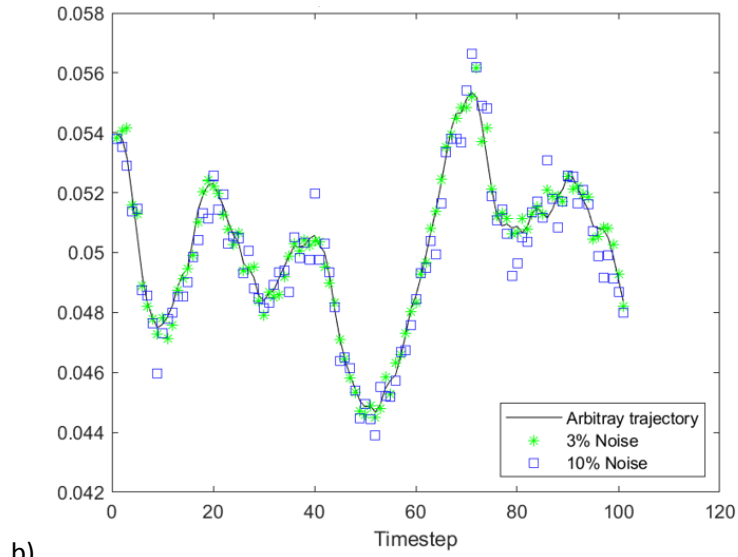


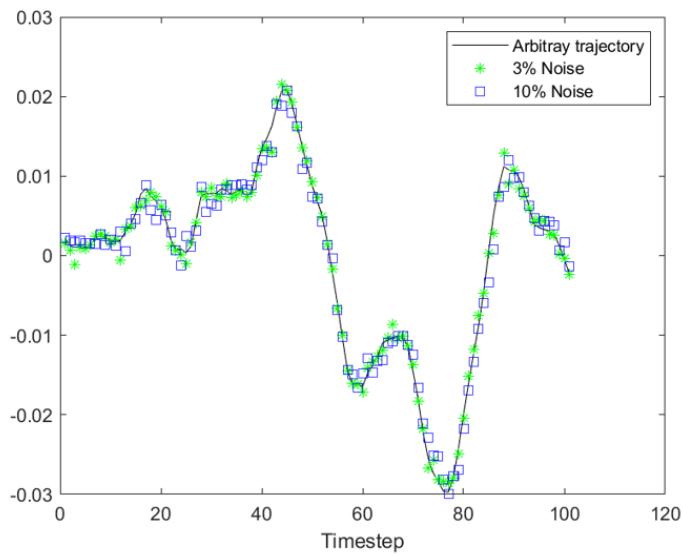
Figure 6 A simulation of a magnetic particle doing an arbitrary translational and rotational motion. The red line stands for the trajectory and the blue arrow represent the orientation of the magnet moment. The boxes stand for the position of the magnetometers.

The arbitrary trajectory with the reconstructed trajectory from the sensor reading containing noises are plotted in Figure 6. Figure 6 a), b) and c) shows the result for x, y and z component respectively. With a 3% of noise in the measurement, MPT method could capture the motion of the particle well. There is a larger deviation at noise level 10% but the trend of the trajectory is obtained. Note that, in the simulation and preliminary test the data processing algorithm is Sequential Quadratic Programming (SQP) method. Because in some previous study, researcher points out that SQP is one of the most suitable algorithms for the magnetic dipole equation. The detail of the algorithm will be discussed in chapter 3.





b)



c)

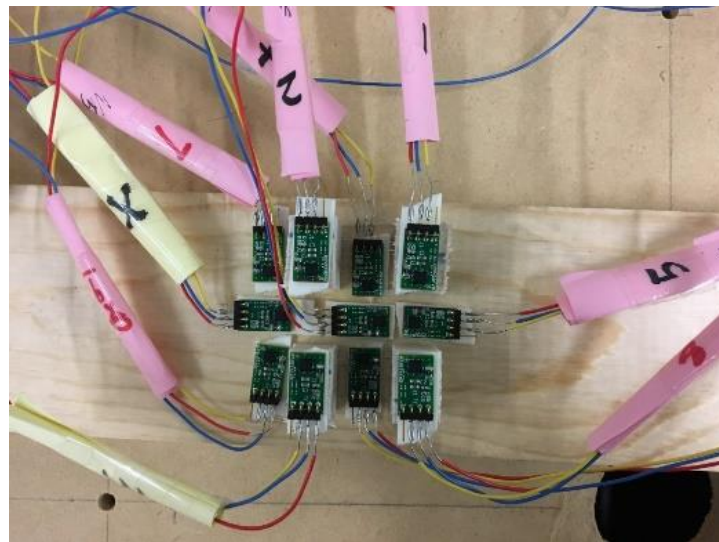
Figure 7 The reconstructed trajectory using MPT method with SQP algorithm in x, y and z directions for a), b) and c).

2.3 Preliminary Experiment

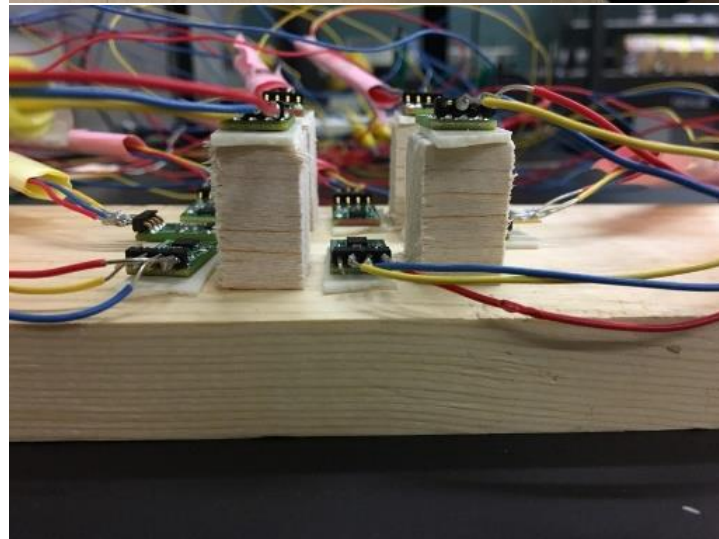
We start preliminary test with low resolution magnet sensors and simple movements. In the preliminary test, the magnet sensors are Texas Instruments DRV 452 and the cost of each is only

\$20. It is a fluxgate sensor with range $[-500 \mu T, 500 \mu T]$. There are total 11 sensors used to form the sensor array as shown in figure 7. Note that to solve the magnetic dipole equation only need 5 sensor measurements, however, the signal decreases with a cubic ratio of distance from magnetic particle to sensors which means the signal to noise ratio (SNR) could be small when the tracer particle is far away from the sensor. So, using more sensors can enlarge the test domain to ensure there will be at least 5 valid measurements to capture the motion of the particle. And the accuracy can be increased by adding more weight to sensors which are at their best working range. All the magnet sensors are installed on a wood plate and no ferromagnetic materials are allowed on the test bench, each sensor is calibrated with their position and numbered to record the data. Before experiment, the empty domain is measured as the offset. Since the response time of the sensor is large and the sampling frequency is not very high, we start with simple motions. The first one we did is a pendulum motion; one magnetic bead is super glued with a string hanging above the sensor array shown as figure 7 c). This is a simple motion as the pendulum is only moving in a plane as a 2D movement. We use magnetometer to record the magnetic field change and the motion of the magnetic bead is also recorded with a high-speed camera. Two sets of trajectories are reconstructed, one is using the MPT technique by solving the magnetic dipole equation based on the sensor readings; the other is using image processing technique to extract and detect the position of the magnetic field. Since, the image-based technique has a very high accuracy, we can take it as the ground truth. For the image-based method, the position of the ball can be directly obtained when the ball is in the view of the camera. The position of the ball can also be calculated by the angle of the thread away from the vertical line when the ball is behind the wood strut. Figure 7 c) shows the reconstructed trajectory using SQP algorithm matches with the position acquired from high speed camera well. This damping sinusoidal curve with a constant frequency is expected as there

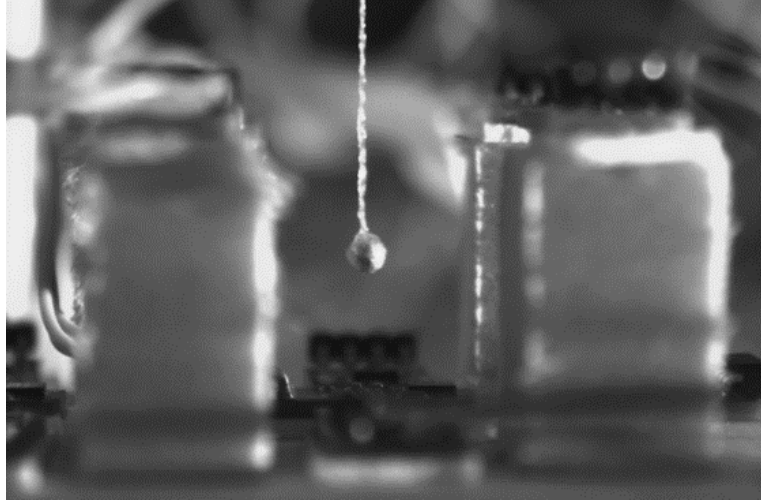
are frictions between the thread and the fix point. The two results match well evident that MPT is a feasible way to tracking the motion of the magnetic particle in the y-z plane. We also notice that there are some discrepancies between the high-speed camera and MPT reconstructed trajectory at the turning point which is due to the hysteresis of the sensor. This issue can be solved by implement more advanced magnetometers. After the success of this preliminary test, we are using Bartington 612 3-axis probe for experiment instead of this Texas Instruments DRV 452.



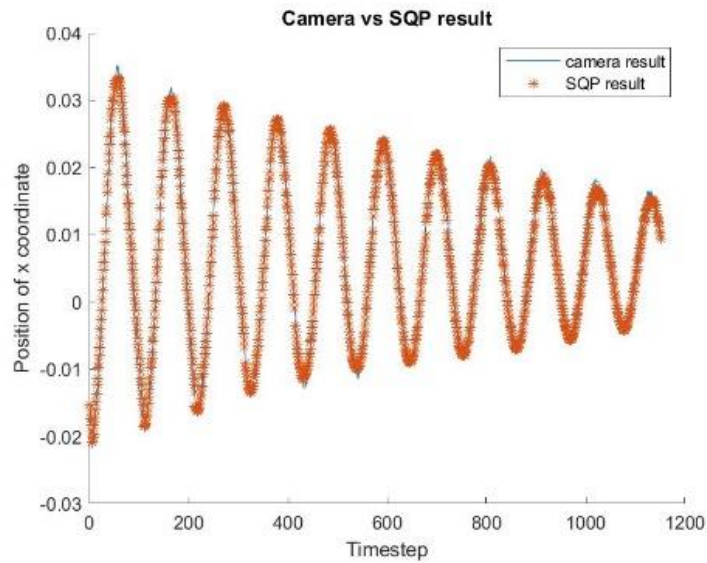
a)



b)



c)



d)

Figure 8 a) and b) are the low-resolution magnet sensor array top view and side view respectively c) is the photo pendulum experiment taking by a high-speed camera d) is comparison of the reconstructed trajectory using SQP with the image based trajectory.

3 Magnetic Particle Tracking Algorithms

3.1 Standard Optimization Method

Previous chapter has shown that a standard optimization method is able to deal with the magnetic dipole equations. Researchers have explored multiple optimization algorithms to addressing this problem. Hu had tried non-linear localization algorithm such as Powell, Downhill, DIRECT, MCS and L-M and found that Powell's method results in a large error regarding the reconstructed position, downhill simplex has a very small tolerance for initial guess, while DIRECT and MCS can provide accuracy but their execution speed is too slow. Levenberg-Marquardt (L-M) method is the appropriate choice because it provides satisfactory tolerance for initial guess, very low computation error and faster speed. They also tried to use linear algorithm because the nonlinear algorithms are low-speed, high complexity and dependence on the initial guess. The magnetic dipole equation can be simplified as a linear form as

$$F_l * \Gamma = b_l \quad (l = 1, 2, 3 \dots, N) \quad (8)$$

Where

$$F_l = [F_{l1}, F_{l2}, F_{l3}, F_{l4}, F_{l5}] = B_{lx}, B_{ly}, B_{lz}, (B_{lz}y_l - B_{ly}z_l), (B_{lx}z_l - B_{lz}x_l)$$

$$\begin{aligned} \Gamma &= [r_1, r_2, r_3, r_4, r_5]^T \\ &= [(b - cn'), (cm' - a), (an' - bm'), m', n']^T \end{aligned}$$

With $r_4 = m' = m/p$, $r_5 = n' = n/p$ and $b_l = B_{lx}y_l - B_{ly}x_l$

In equation (6), F_l and b_l are composed of the sensor data (B_{lx}, B_{ly}, B_{lz}) and its position (x_l, y_l, z_l) , and independent of the six unknown parameters (a, b, c, m, n, p) ; With the data from

five sensors, a matrix $M = [F_1, F_2, F_3, F_4, F_5]^T$ and a vector $B = [b_1, b_2, b_3, b_4, b]^T$ can be calculated and then we have

$$\Gamma = M^{-1}B.$$

Once Γ is solved, then all the parameters can further be solved. Through the experiments on the localization system, they found that nonlinear algorithm provides better accuracy but takes more computing time while linear algorithm is faster but with a larger error. So, they combined these two methods together, used the linear algorithm to find the localization and orientation parameters and then the nonlinear algorithm is applied for further computation by using the initial parameters obtained from the linear algorithm. Since, the initial parameter obtained from linear algorithm is close to globally true position the nonlinear algorithm converges fast to the correct solution. They also plotted the error against sensor number, the error drops significantly from sensor number 5 to 10 and then gradually converges to 1.8 mm in position and 1.54°.

However, Buist points out for research purposes, the algorithm does not need to be fast and investigates different optimization algorithms. They use 72 sensor signals and use the bound of the sensor array as the constraint. To determine the position and orientation of the tracer particle, solving the magnetic dipole equation becomes minimizing a quality function which is the difference between the theoretical signal strengths to the actual signals. The quality function is given in equation 9

$$Q = \sum_{i=1}^{72} \frac{((S_{m,i} - \langle S_m \rangle) - (S_{t,i} - \langle S_t \rangle))^2}{\Delta S_{m,i}^2} \quad (9)$$

The quality function 9 is corrected for deviations in each individual sensor given by a min/max deviation $\frac{1}{\Delta S_{m,i}^2}$, this ensures that less accurate sensors have less influence on the quality function.

Buist further points out that in the L-M algorithm, the solution is done by solving for the position x , y , z and the angles φ , θ and due to the periodicity of φ , it is difficult to solve and result in erroneous solutions. However, the Sequential Quadratic Programming (SQP) algorithm can add one degree of freedom by changing the orientation unit vector and adding a nonlinear constraint for the norm of the vector shown as in equation 10.

$$\begin{aligned}
 |\bar{e}_p| &= 1 & (10) \\
 -1 &\leq e_x \leq 1 \\
 -1 &\leq e_y \leq 1 \\
 -1 &\leq e_z \leq 1
 \end{aligned}$$

We had also tried multiple optimization methods and confirmed that the SQP is one of the best among all candidate choices. However, the state-of-the-art MPT technology has a severe limitation: its reconstruction process is too slow. The reconstruction is an optimization process of finding the magnet's position and orientation. The cost function in the optimization is highly nonlinear and possesses many local minimums. Hence the reconstruction, which aims to find the global minimum is time consuming.

3.2 Particle Filter

If we take another look of this problem, the tracking process could be an estimating of the state of a dynamic system when partial observations are made, and random perturbations are present in the sensors as well as in the dynamic system. Then the Particle filters or Sequential Monte Carlo (SMC) methods are good candidates for solving this problem. Particle filter is a set of Monte Carlo algorithms, it uses a set of particles or samples to represent the posterior distribution of some stochastic process given noisy and/or partial observations. The state-space model can be nonlinear

and noise distribution can take any form required. Particle filter techniques provide a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model.[72], [73] One shortage of Particle filter is that it does not perform well when applied to very high-dimensional systems. But since we are only tracking one particle in the system, so the dimension of this problem is not high. Therefore, particle filter may a good choice for the magnetic particle tracking.

Particle filter update its prediction in a statistical manner. The samples from the distribution are represented by a set of particles, each particle has a likelihood weight assign to it which represent the probability the particle being sampled from the probability density function. The weight disparity issue could be mitigated by resampling step. Here in our project, the algorithm is Sequential Importance Sampling with resampling (SIR[74], [75]) .

The mathematical foundations and rigorous analysis of particle filter is done by Pierre Del Moral in 1996. He proved that the unbiased properties of a particle approximations of likelihood functions and the unbiased particle estimator of the likelihood function is still used today in Bayesian statistical inference.

A generic particle filter estimates the posterior distribution of the hidden states by the observation shown as Figure 9

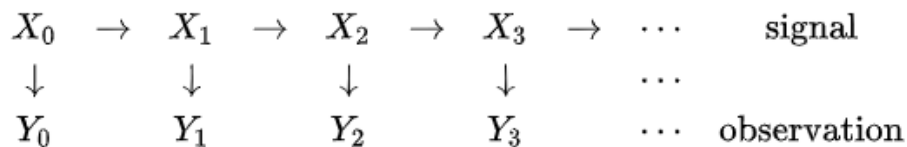


Figure 9 a state-space diagram

The filtering problem is to estimate sequentially the values of X_k , given the values of the observation Y_0, \dots, Y_k at any time step k .

A standard particle filter process is shown as figure 10. Suppose at time step $k - 1$, we have presented $p(\mathbf{X}_{0:k-1} | O_{1:k-1})$ by a set of samples $\{\mathbf{X}_{0:k-1}^i\}_{i=1}^{N_s}$ with the associated weights $\{\omega_{0:k-1}^i\}_{i=1}^{N_s}$, here N_s is the number of virtual samples. In this paper, 5000 samples are used in the MPT reconstruction. At time step k , we first use the model equation (4) to propagate the samples to step k as $i = 1, \dots, N_s$,

$$\mathbf{X}_k^i = \begin{pmatrix} r_k^i \\ m_k^i \end{pmatrix} = F(\mathbf{X}_{k-1}^i) + w_k^i \quad (11)$$

where w_k^i is a realization of the Gaussian variable w_k with a zero mean and a covariance Q_k . The estimation of \mathbf{X}_k at step k is approximated by the weighted mean of the samples, i.e., $\mathbf{X}_k = \sum_{i=1}^{N_s} \omega_k^i \mathbf{X}_k^i$. The uncertainty is approximated by the weighted sample covariance.

The observation equation:

$$O_{j,k} = D_j(\mathbf{X}_k)(1 + \mathbf{v}_{j,k}) \quad (12)$$

is used to calculate the likelihood $P(O_k | X_k^i)$. where $\mathbf{v}_{j,k}$ is a gaussian random noise and $D_j(\mathbf{X}_k)$ is the theoretical measurement of the sensor

$$D(\mathbf{X}_k) = \begin{pmatrix} D_1(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_1 \\ \vdots \\ D_j(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_j \\ \vdots \\ D_N(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_N \end{pmatrix} \quad (13)$$

Again here, the S_j is the orientation of the sensor since the sensor typically measures only one specific direction. And the weight for each virtual sample is updated as

$$\omega_k^i = \omega_{k-1}^i P(O_k|X_k^i) \quad (14)$$

Finally, these weights are normalized, as their sum should be 1.



Figure 10 a standard particle filter process

If the variance of the weights is too large, resampling is used to make the weight uniform (see the algorithm 3 in Ref [76] for details, here we gives a screenshot of the pseudo code in Figure 11).

```

Algorithm 3: Generic Particle Filter
[{\mathbf{x}_k^i, w_k^i}_{i=1}^{N_s}] = PF[{\mathbf{x}_{k-1}^i, w_{k-1}^i}_{i=1}^{N_s}, \mathbf{z}_k]
• FOR i = 1: N_s
  - Draw \mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)
  - Assign the particle a weight, w_k^i,
    according to ( )
• END FOR
• Calculate total weight: t = SUM[{\omega_k^i}_{i=1}^{N_s}]
• FOR i = 1: N_s
  - Normalize: w_k^i = t^{-1} \omega_k^i
• END FOR
• Calculate \widehat{N}_{eff} using (15)
• IF \widehat{N}_{eff} < N_T
  - Resample using algorithm 2:
  * [{\mathbf{x}_k^i, w_k^i, -}_{i=1}^{N_s}] = RESAMPLE[{\mathbf{x}_k^i, w_k^i}_{i=1}^{N_s}]
• END IF

```

Figure 11 Pseudo code of SIR

The effective sample size is:

$$N_{eff} = \frac{N}{1 + \text{Var}(w_i^{*n})} \approx \frac{1}{\sum_{i=1}^N (w_i^n)^2} \quad (15)$$

Where N is the number of particles w_i^{*n} is the true weights and w^n is the normalized weights. If the effective number of particles is less than a given threshold, then the performance of resampling will start to work:

- a) Draw N particles from the current particle set with probabilities proportional to their weights.
Replace the current particle set with this new one.
- b) For $i = 1, \dots, N$ set $w_k^i = 1/N$

The resampling process could get rid of the particles with small weights and preserve the probability density. Namely, each particle has the same weight afterwards and only few particles left to present the posterior density. Figure 12 shows the process of resampling, the red dot represents the sampling points which has very little probability to be sampled therefore they are eliminated. And the upper big green dots are resampled to have multiple equal weights particles.

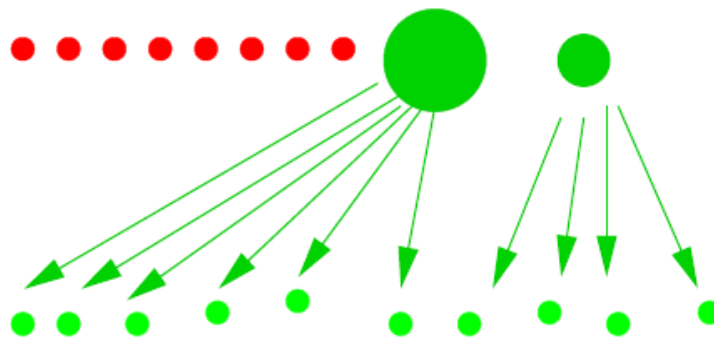


Figure 12 the process of resampling where red dots are eliminated while the upper big green dots become multiple equal weight particle after resampling

3.3 Extended Kalman Filter

Kalman filter also known as linear quadratic estimation (LQE) is an algorithm produces estimations of unknown variables that tend to be more accurate than those based on a single measurement alone by estimating a joint probability distribution over the variable for each time frame. It can trace back to 1960 and has a broad application in statistic and control theory, it is named after Rudolf E. Kalman. It is first being used in the implementation of the navigation of navy ballistic missile. It uses the dynamic model of a system (usually based on physical laws of motion), known control inputs to that system and multiple sequential measurements (from sensors) to form an estimate of the system's varying state. Since it uses more sensor measurements instead one, it has a better performance on the state estimate or predictions.

The Kalman filter produces the estimate of state of a system as an average of the system's predicted state and of the new measurement using a weighted average. The better (i.e., smaller) estimated uncertainty the higher value of the weight it will be given. The weights are calculated from covariance which measures the estimated prediction's uncertainty of the state of system. The result of the weighted average is a new state estimate that sits between the predicted state and the measure one, and it has a better estimated uncertainty than either one. This process is repeated at each time step and this recursively work only requires the last best guess rather than the entire history.

The diagram of how Kalman filter to estimate the internal state of a process given a sequence of observation containing noise is shown in figure 13 (this figure is from Wikipedia). The definition of each is listed as below:

- \mathbf{F}_k , the state-transition model;
- \mathbf{H}_k , the observation model;
- \mathbf{Q}_k , the covariance of the process noise;
- \mathbf{R}_k , the covariance of the observation noise;
- and sometimes \mathbf{B}_k , the control-input model, for each time-step, k , as described below.

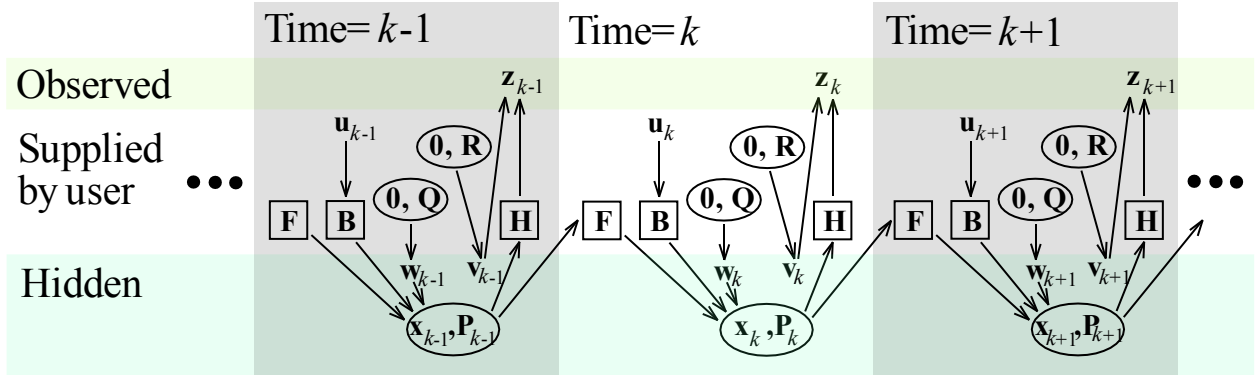


Figure 13. Model underlying the Kalman filter. Squares represent matrices. Ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors. In the simple case, the various matrices are constant with time, and thus the subscripts are dropped, but the Kalman filter allows any of them to change each time step.

In the MPT problem Let the state variable $\mathbf{X}_k = \begin{pmatrix} r_k \\ m_k \end{pmatrix}$ be the particle status at time step k . Given \mathbf{X}_{k-1} , \mathbf{X}_k can be modeled as follows: The location of the magnetic particle r_k depends on its position and velocity at previous step $k - 1$ and can be modeled by a kinematic equation

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{u}_{k-1}\Delta t + \mathbf{w}_k^r \quad (16)$$

The rotation of the magnetic particle can be calculated using a quaternion q as

$$\mathbf{m}_k = \mathbf{q}_{k-1}\mathbf{m}_{k-1}\mathbf{q}_{k-1}^{-1} + \mathbf{w}_k^m \quad (17)$$

here \mathbf{u}_{k-1} is the velocity of the magnet at time $k - 1$, Δt is the time interval, \mathbf{w}_k^r and \mathbf{w}_k^m are two independent Gaussian variables, which represent unknown perturbations or random noises in the motion and rotation equations, and $q = (\cos \frac{\Omega \Delta t}{2}, \omega_x \sin \frac{\Omega \Delta t}{2}, \omega_y \sin \frac{\Omega \Delta t}{2}, \omega_z \sin \frac{\Omega \Delta t}{2})$. The $(\omega_x, \omega_y, \omega_z)$ is a unit vector along the axis of rotation and Ω is the magnitude of angular velocity. The \mathbf{u} and q can be estimated using historic trajectory data. Therefore, we obtain our forward model of the state variable \mathbf{X}_k as

$$\mathbf{X}_k = \begin{pmatrix} \mathbf{r}_k \\ \mathbf{m}_k \end{pmatrix} = F(\mathbf{X}_{k-1}) + \mathbf{w}_k \quad (18)$$

where

$$F(\mathbf{X}_{k-1}) = \begin{pmatrix} \mathbf{r}_{k-1} + \mathbf{u}_{k-1} \Delta t \\ \mathbf{q}_{k-1} \mathbf{m}_{k-1} \mathbf{q}_{k-1}^{-1} \end{pmatrix},$$

and

$$\mathbf{w}_k = \begin{pmatrix} \mathbf{w}_k^r \\ \mathbf{w}_k^m \end{pmatrix}.$$

This is similar to the particle filter and so does the observation equation

$$O_{j,k} = D_j(\mathbf{X}_k)(1 + \mathbf{v}_{j,k}) \quad (19)$$

Where

$$D(\mathbf{X}_k) = \begin{pmatrix} D_1(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_1 \\ \vdots \\ D_j(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_j \\ \vdots \\ D_N(\mathbf{X}_k) = B(\mathbf{X}_k) \cdot S_N \end{pmatrix} \text{ and } \mathbf{v}_k \text{ is a vector of Gaussian random variables, which are}$$

independent of \mathbf{w}_k and represent the measurement uncertainties. We assume that \mathbf{v}_k has a zero mean and the covariance matrix is R_k .

Equations (16) and (17) together form a state-space model. The task of MPT reconstruction is equivalent to estimating the state variable \mathbf{X}_k based on the measurements from time step 1 to k $O_{1:k}$. The conditional probability distribution of the state \mathbf{X}_k given the measurement $O_{1:k}$, is denoted by $p(\mathbf{X}_k|O_{1:k})$. The mean of $p(\mathbf{X}_k|O_{1:k})$ will give the minimum mean square error estimate of the state \mathbf{X}_k and the uncertainty is given by the covariance of $p(\mathbf{X}_k|O_{1:k})$.

In the EKF, $p(\mathbf{X}_k|O_{1:k})$ is approximated by a Gaussian distribution, which can be completely characterized by the mean and covariance matrix. We assume the mean at time step $k - 1$ is \mathbf{X}_{k-1} and the covariance is P_{k-1} . Subsequently, the mean \mathbf{X}_k and the covariance P_k are obtained in two steps: prediction and analysis.

(a) In the prediction step, the predicted mean of the trajectory \mathbf{X}_k^- is obtained using the model equation (16)

$$\mathbf{X}_k^- = \begin{pmatrix} r_k^- \\ m_k^- \end{pmatrix} = F(\mathbf{X}_{k-1}) = \begin{pmatrix} r_{k-1} + \mathbf{u}_{k-1}\Delta t \\ \mathbf{q}_{k-1} \mathbf{m}_{k-1} \mathbf{q}_{k-1}^- \end{pmatrix} \quad (20)$$

The covariance matrix P_{k-1} is updated to P_k^- as

$$P_k^- = F_k P_{k-1} F_k^T + Q_k$$

where $F_k = \frac{\partial F}{\partial \mathbf{X}}(\mathbf{X}_{k-1})$ is the Jacobian of the model function F evaluated at \mathbf{X}_{k-1} .

(b) In the analysis step, O_k , the measurement at time step k , is used to update \mathbf{X}_k^- and P_k^- by the Kalman formula as follows

$$\mathbf{X}_k = \mathbf{X}_k^- + K_k(O_k - D(\mathbf{X}_k^-)) \quad (21)$$

$$P_k = (I - K_k H_k) P_k^- \quad (22)$$

where the matrix K_k is the standard EKF correction matrix as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}. \quad (23)$$

Here $H_k = \frac{\partial D}{\partial X}(\mathbf{X}_k^-)$ is the Jacobian matrix of the measurement function D evaluated at \mathbf{X}_k^- , and

$V_k = \frac{\partial O_k}{\partial v_k}$ is the Jacobian of the observation with respect to \mathbf{v}_k .

3.4 Comparison of Algorithms

In this paper, we test Sequential Quadratic Programming (SQP), extended Kalman filter (EKF [77], [78]) and Sequential Importance sampling with Resampling (SIR [76], [79]). The latter two methods are based on the reformulation of the MPT as a state space model, which is widely used in science and engineering to identify the state of a system from an uncertain model supplemented by a stream of noisy data. Because their performances depend on the specific problem, we test both of them and compare with the SQP method to identify the best approach. In the state space model, the state of the system is regarded as a stochastic process, and its conditional probability distribution based on the given data is approximated. The conditional mean will give the minimum mean square error estimation of the state, and the uncertainty is given by the covariance. If the model and the observation function are both linear and the noises (in the model and observation) are Gaussian, the conditional mean and covariance can be calculated by the Kalman filter [80], [81]. The EKF is an extension of the Kalman filter. It is capable of handling complex dynamics and nonlinear observation functions, but the noises in both model and observation are still assumed to be Gaussian. The SIR is one of the particle filters [76], [79], [82], which can be used for general nonlinear non-Gaussian situations. We compare the results obtained using the EKF and SIR with those from the SQP. The numerical experiments show that the EKF provides the same accuracy as the SQP, but it is orders of magnitude faster. The efficiency of the SIR falls between the EKF and SQP.

We use several simulations to evaluate the performances of the SQP, SIR, and EKF methods in the reconstruction of trajectories. Not to lose generality, a synthetic trajectory is created by randomly perturbing the motion of an object so that the velocity becomes discontinuous along the trajectory (Figure 2). The magnet moment is set to be 0.0105 Am². A real magnet with this moment is used in the following experiments. The blue curve indicates the location of the magnet, and the red quivers represent the magnet moment. To evaluate the robustness of SQP, SIR, and EKF at multiple noise levels, Gaussian noise is added to the synthetic sensor readings. Specifically, for the i -th sensor at each time step, O_i is set to be $\mathbf{B}(\mathbf{r}, \mathbf{r}_{0i}, \mathbf{m}) \cdot \mathbf{S}_i \cdot (1 + \epsilon)$, where ϵ has a normal distribution with a zero mean. The standard deviation of ϵ is selected to be 0.01, 0.03, 0.06, 0.1 and 0.2 in different simulations. Multiple trajectories are reconstructed based on the data with these noise levels. Figure 3 shows a comparison of the synthetic and reconstructed trajectories. The total time step in the synthetic trajectory is 5,000, but only a section (with no noise and 3% added noise) is presented for clarity.

To set a standard in estimating the accuracy of MPT reconstruction, we define the relative error as the difference between the real and reconstructed positions divided by the domain size. For the x component, $E_x = \text{mean}(|x_j^* - x_j|)/L_x$. Here, L_x is the size of the domain in the x direction, $L_x = [\max(x) - \min(x)]$, x_j^* is the real value of the position. In the simulation, it is the position of the synthetic trajectory. Similarly, we can calculate the errors in the y and z directions. Consequently, the average relative position error of all components is $\frac{1}{3}(E_x + E_y + E_z)$. The orientation error can be measured using the mean angle between the measured vector \mathbf{m} and the real \mathbf{m}^* .

Table 2 shows the uncertainties in position and direction obtained from the above simulations, and the same results are plotted in Fig. 16. We notice that the SIR may not return a useful value when the noise level is too high. The noise level of a typical magnetometer is smaller than 5%. Taking the 3% noise case as an example, the relative position error of EKF results is 0.52%, better than the SQP and SIR. In the lower noise cases, the SQP is slightly better. With 3% noise, the angular accuracy of MPT is in the order of 1 degree. The SQP orientation result is better than the other two approaches. We can compare the position errors with that of optical methods. The image-based position uncertainty can be estimated as 1 pixel and the field of view is typically 1K×1K. Hence, the relative error is in the order of 0.1%. Therefore, the accuracy of MPT position reconstruction is not very far from that of optical methods.

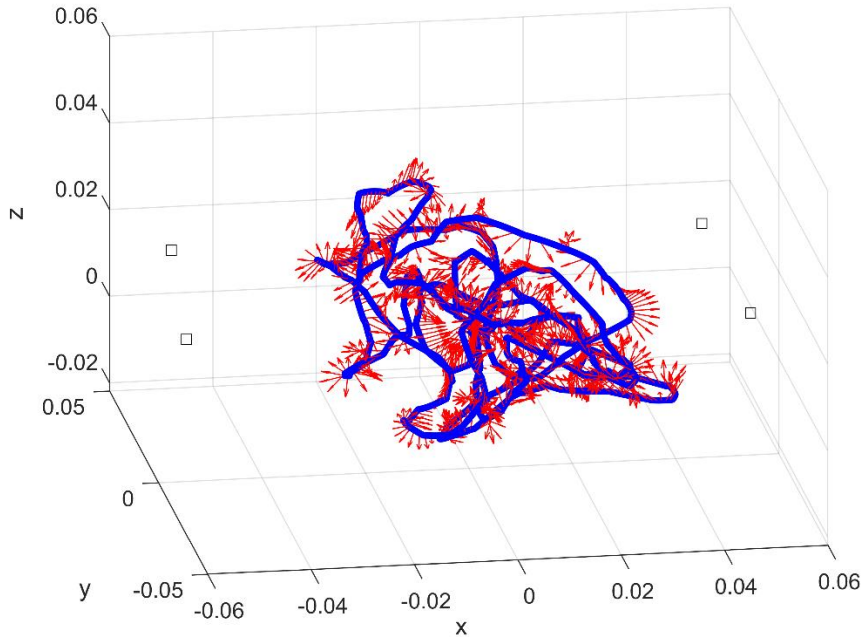
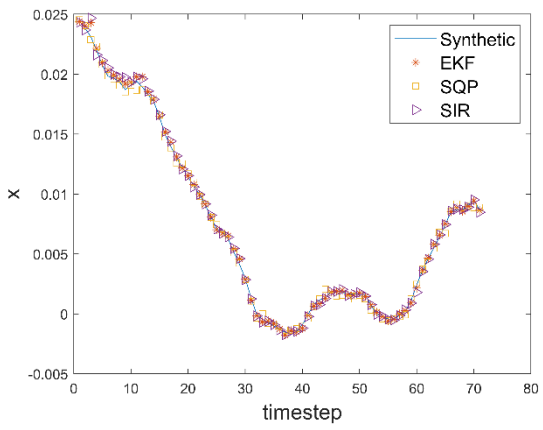
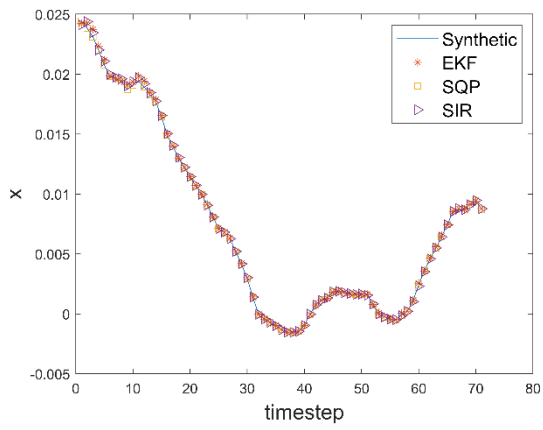
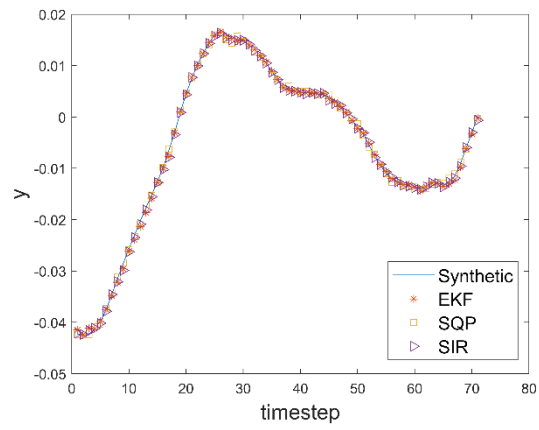
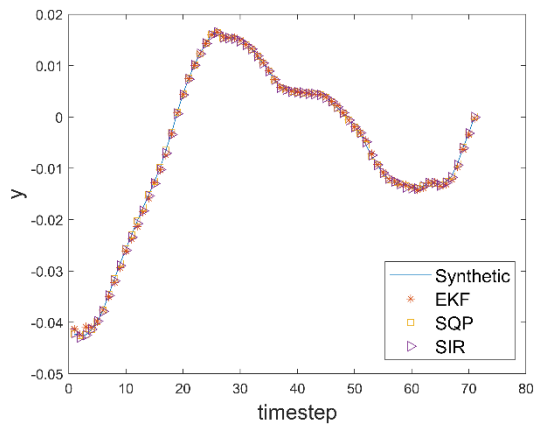


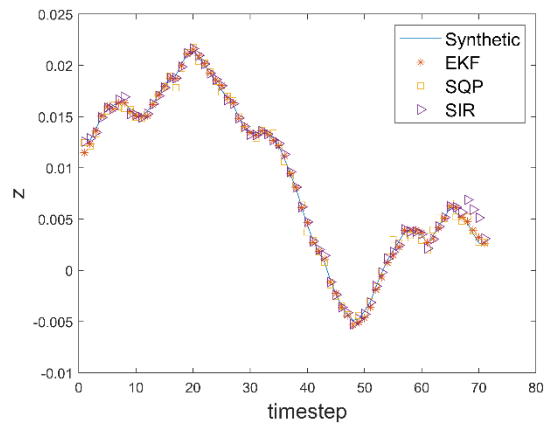
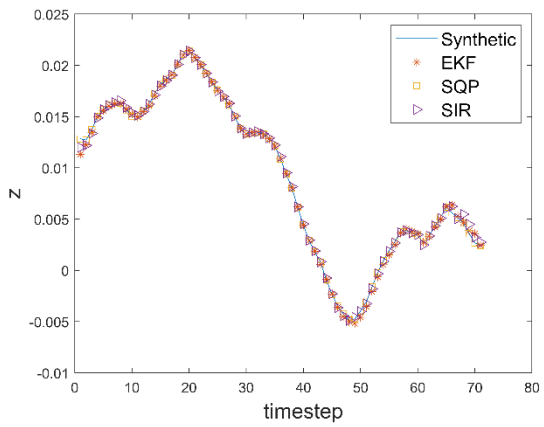
Figure 14. The blue path is the synthetic trajectory of a magnetic ball and the red arrows represent the orientation of the magnet. The square boxes represent the magnetometers in the simulations. The unit of the axis is meter.



a)



b)



c)

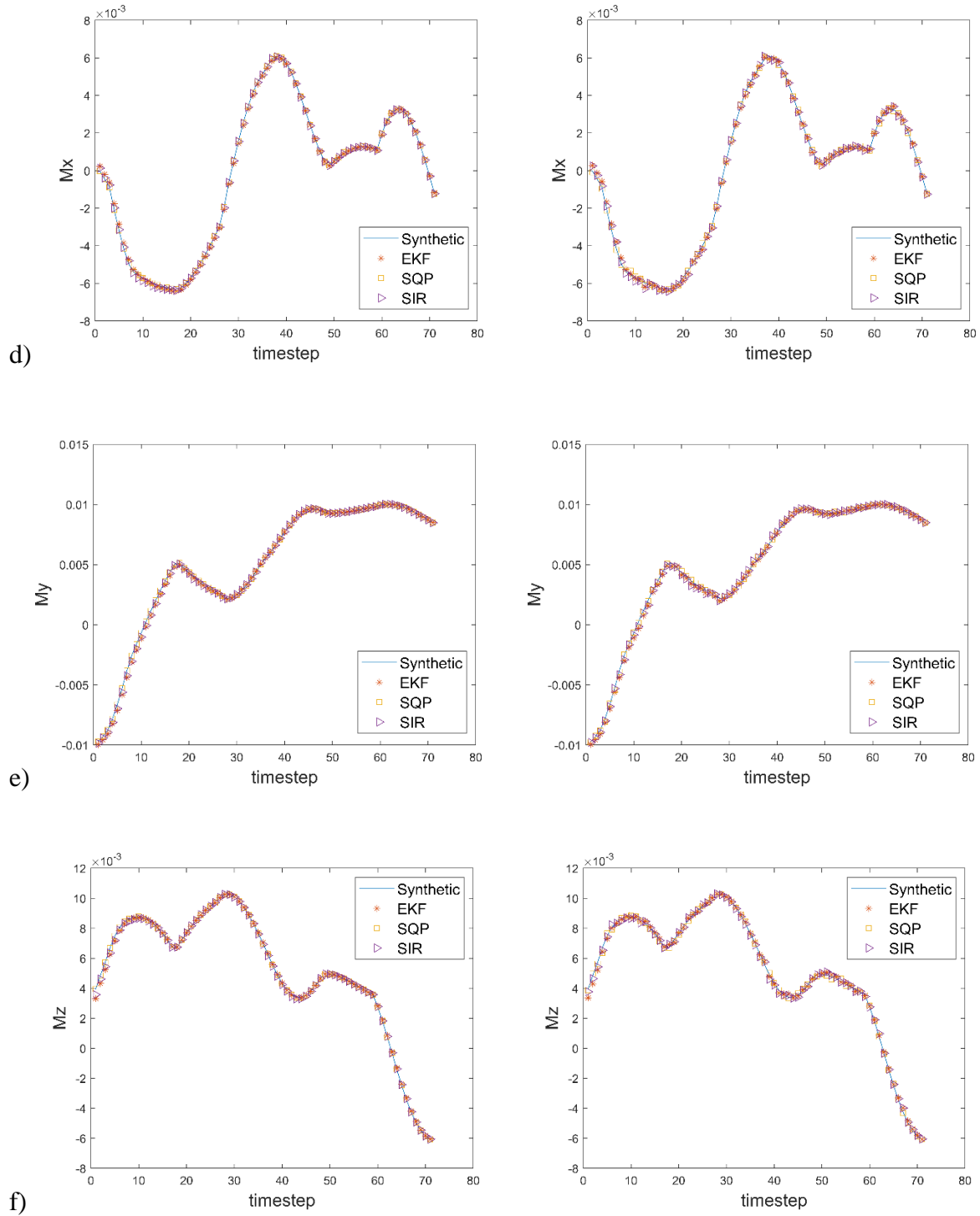


Figure 15. Sample trajectory sections reconstructed using the SQP, SIR and EKF with no noise (the left column) and 3% added noise (the right column). a), b) and c) are for x, y, and

z component respectively; d), e) and f) are for M_x , M_y and M_z respectively. The unit of x , y and z is meter and that of M_x , M_y and M_z is Am^2 .

Table 2 The position and rotation error versus the level of noise

Error	Method	No noise	1%	3%	6%	10%	20%
Position error	SQP	0.003%	0.21%	0.6%	1.1%	1.49%	8.41%
	EKF	0.24%	0.3%	0.52%	0.94%	1.51%	9.88%
	SIR*	0.41%	0.5%	0.86%	1.56%	–	–
Rotation error (degree)	SQP	0.053	0.30	0.83	1.61	2.6	5.73
	EKF	0.76	0.81	1.10	1.78	2.7	5.54
	SIR*	1.43	1.43	1.74	2.77	–	–

*In the SIR computation, 5000 samples were used.

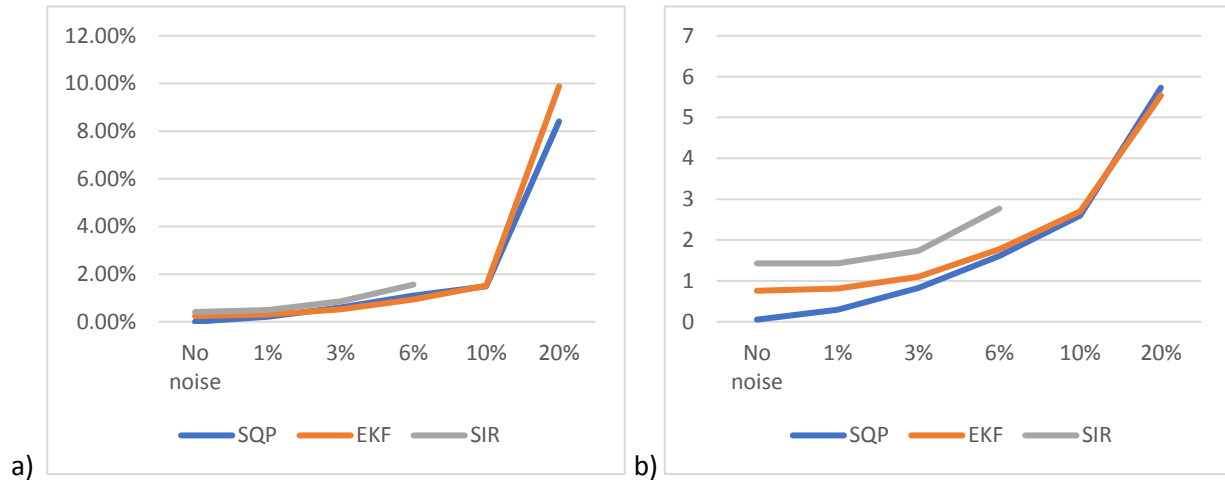


Figure 16. Position error dependence on the noise, and b) The rotation error versus the noise level.

Despite the above methods having similar robustness against noise, their processing speeds differ significantly. In our simulation test, SQP algorithm takes about 2 hours to proceeding data, while the SIR takes about one third the time of the SQP; and the EKF only takes 10 seconds. The

reason is that the SQP solves multiple sub-optimization problems without the guidance of a model equation, i.e., it breaks one-time step into multiple small steps. In contrast, the EKF directly estimates the most probable position of one particle in one step. As for the SIR method, since it generates and follows thousands of virtual samples, its speed is much slower than the EKF. Here, we want to mention that to compare each algorithm under a fair environment, no parameters have been changed in any of the algorithms. This is actually slightly ‘unfair’ for SIR algorithm, since the scope of the virtual sample is fixed if no parameters are changed while the increasing with noise level will leading to a misleading direction for SIR. But this situation is better for EKF algorithm, since it uses a Gaussian distribution to estimate the noise. Considering both accuracy and speed, the most cost-effective method is the EKF. Therefore, we use it to process our data in the following experiments.

4 Experiment and Data Analysis

4.1 A comparison Experiment of MPT Method with High Speed Camera

A floating magnet is used to experimentally evaluate the MPT uncertainty. The magnetic particle is made of neodymium rare earth with a diameter of 3 mm. As shown in Fig 17, a plastic bottle cap containing the magnetic ball is floating on the surface of water. To ensure the background light will not have a strong effect on the detection of the magnetic ball using image processing method, a small piece of black marker is covering the magnetic ball at the near center place. And another bigger black marker near the curb of the bottle cap is used to indicate the orientation of the magnet (the angle θ). The cap is driven by random disturbances. In order to obtain an independent

measurement of the motion, a high-speed camera is used to record the magnet's trajectory from the top at a framerate of 1000. We can identify the magnet position and orientation through processing the images. Four Bartington M612 probes (three-axis fluxgate magnetometers) are distributed around the measurement domain. Their sensing range is $[-90\mu\text{T}, 90\mu\text{T}]$. The sampling frequency is set to be 1000 Hz and synchronized with the camera. The coordinates of each sensor are precisely determined by an accurately machined holder. At the beginning of each experiment, the empty domain without any magnet is measured as the offset. And the calibration of the position of the sensor is performed with high speed camera. Also, the camera CMOS sensor pixel converting rate to real distance is determined as well. Thereafter, the magnetic ball is introduced into the domain, where the sensor array measures the magnetic field change. All devices are fixed on a wooden table and ferromagnetic materials are kept away from the workbench. The magnet sensor noise level is about 3% of the measured value.



a)

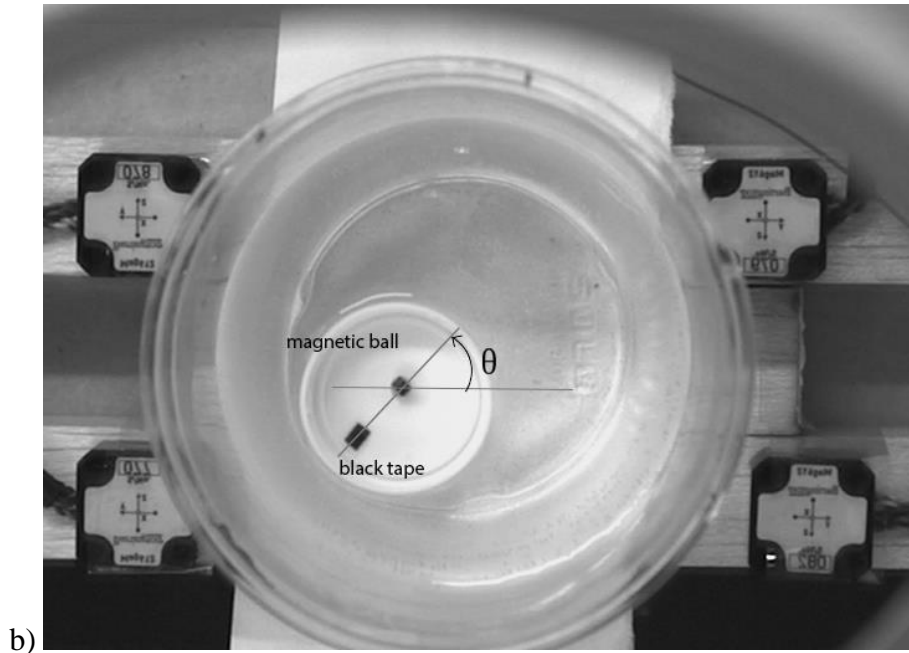
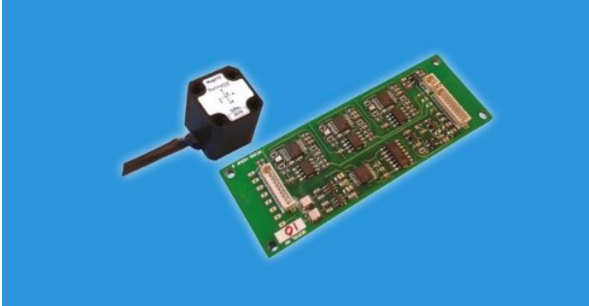




Figure 17. a) The calibration of the sensor in a high-speed camera view. This could convert the pixel on the camera CMOS sensors to real distance. Once the calibration and recording the offset of the test domain finished, the magnet particle is introduced to the test domain. b) A top view of the validation experiment. A cap is floating on the water contained in a plastic bottle. The magnetic ball is glued to the cap center. A black marker near the edge of the cap is used to indicate the orientation of the magnet. Four 3-axis magnetometers are placed near the measurement domain.

Main experiment devices are listed as below:

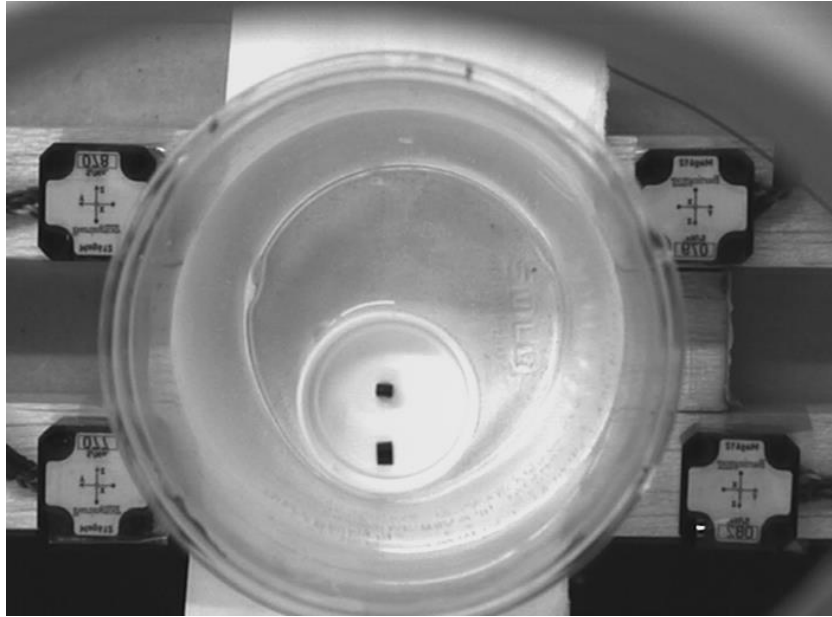
Table 3 Experiment devices and models of the floating cap experiment

Name	Model	Photos	Settings

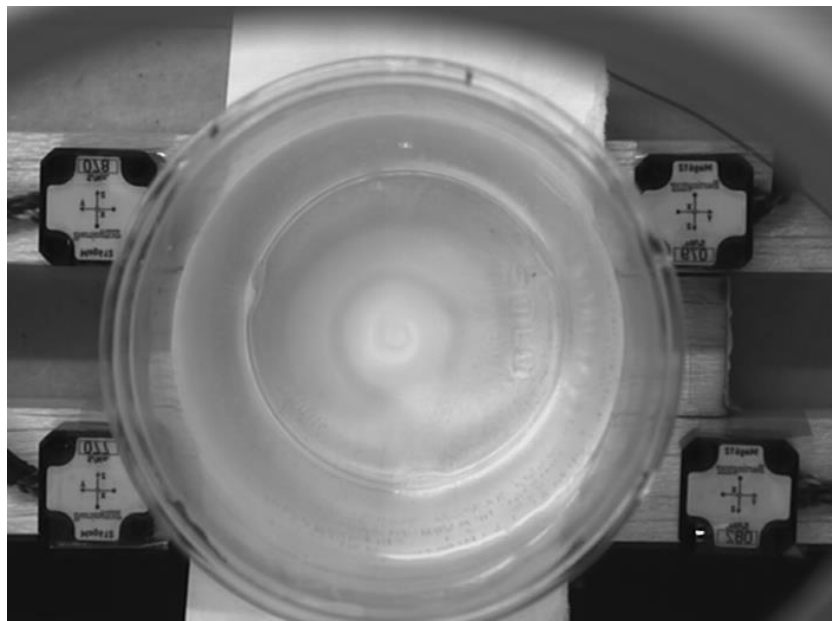
Magnetometer	Bartington 612 × 4		Range: [–90μT, 90μT]; Bandwidth: -3dB; Scaling: 89mV/μT
DAQ	National Instrument USB-6218		Number of Channels: 16; Sample Rate: 250kS/s; Resolution: 16 bits
Camera	nac MEMRECAM HX-7s		CMOS Sensor: 2560 X 1920; Electronic Shutter: 10msec to 1.1usec

This experiment contains both translational and rotational motion of the magnetic particle which is what we want to validate. The bound of the magnetic ball is well under control, in this way, the outrage problem will not happen. The trajectory obtained from camera using image processing technique could be used as the ground truth for comparison of the reconstructed trajectory using MPT methods. Here in figure 18 shows the process of the image processing obtaining the position and orientation information. First, we sum up all the images and doing an

average to extract the background picture shown as figure 18 b), so only stationary part will remain on this background image.



a)



b)



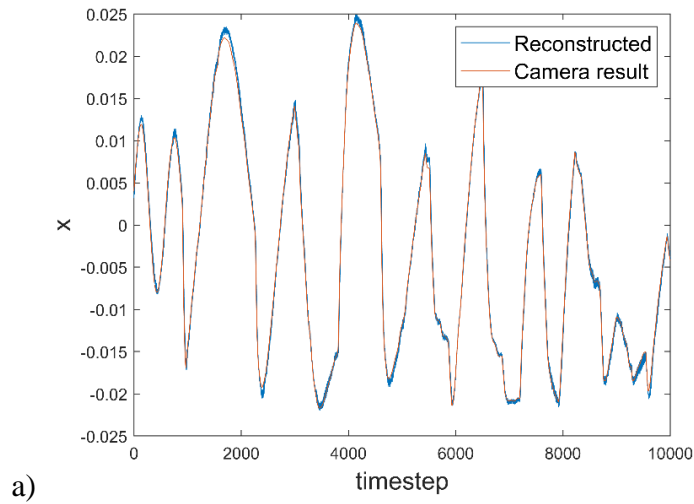
c)

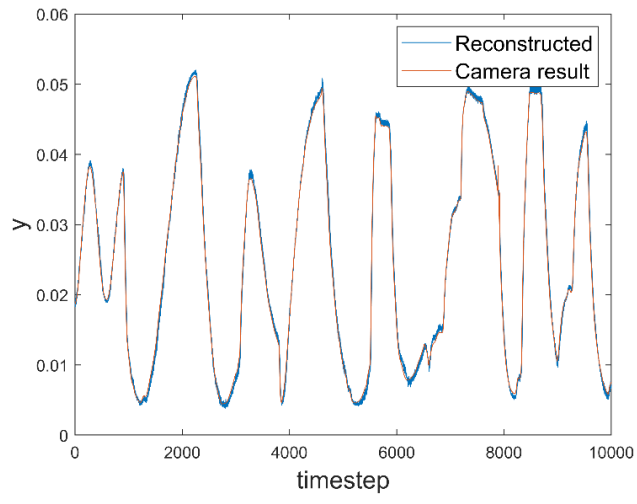
Figure 18. The process of Image Processing to obtain the position and orientation of the magnetic particle a) the original photo at time step k. b) Sum up all the images and doing average to get the background image of this floating cap c) Post-processing images cropped with only the test field and two tappers are identified with star and square markers.

Each original image will subtract the background first and cropped to only test domain to save the storage and fastening the speed of processing. Then an exponential filter will be applied to enhance the contrast of the image. The grey scale of the water and white cap are very high, they can be easily distinguished from the two tappers by setting a threshold. Then a MATLAB “bwlabel” function is used to detect the area of those two tappers, and the area of the marker near the curb is significantly larger than the central one, we can use the size of the area to distinguish each of them. The location of two markers are obtained by averaging the location of pixels of two areas on the

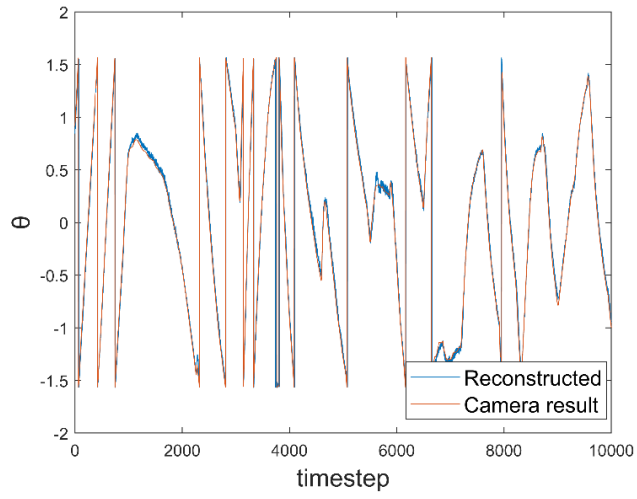
image. Once the location of these two markers known, getting the rotation information is straightforward.

Figure 19 shows that the MPT reconstructed trajectory and the camera-based results almost overlap. Note that the range of rotation angle is $-\pi/2$ to $\pi/2$, so there are sudden jumps between the upper and lower limits in the θ trajectory. Since the image-based results have higher accuracy, we use them as the ground truth to evaluate the uncertainty of the MPT with EKF reconstruction. The average relative position error $\frac{1}{2}(E_x + E_y)$ is 0.65%, and the error in θ determination is 1.44 degree. These results show that the MPT method successfully captures the motion of the tracer particle, which is consistent with the simulation results discussed in the previous section.

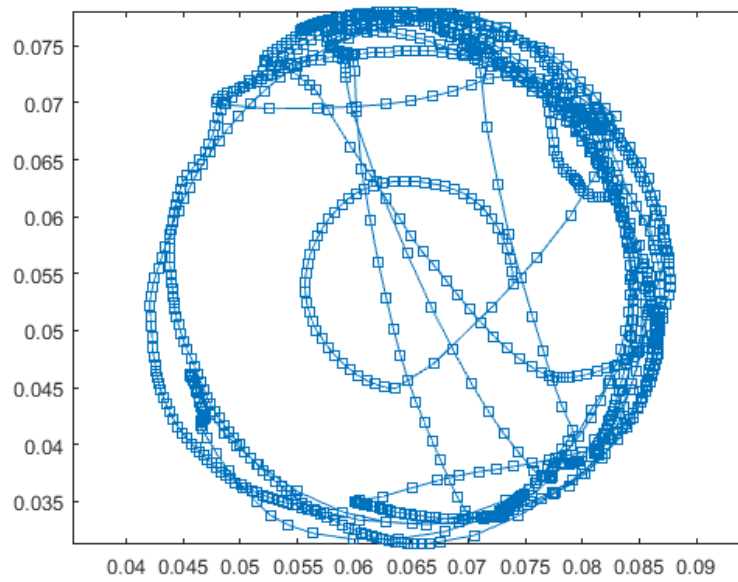




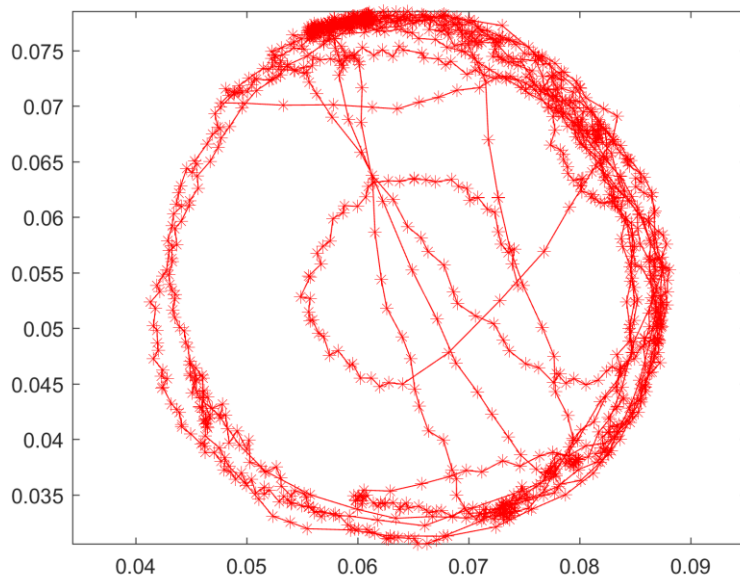
b)



c)



d)



e)

Figure 19. The comparison between the reconstructed trajectory and the result from a camera for a) the x component, b) the y component, and c) rotation angle. d) Trajectory reconstructed by Image processing. e) Trajectory reconstructed using MPT technique with EKF algorithm. The units in a), b), d), e) are meter, and θ is measured in radian.

4.2 Error and Accuracy Analysis

From Figure 19 d) and e), we know that MPT technique with EKF algorithm successfully reconstructed the trajectory of the magnetic particle on the floating cap but for the reconstructed one the path is not as smooth as the image based one. To evaluate the error and accuracy of the velocity using the reconstructed trajectory, we applied MPT technique to capture a particle with controlled motion. One magnetic ball is super glued on a rotating disk driven by a speed-controlled step motor shown as figure 19 a). Since the distance of the magnetic ball to the rotating axis is measurable and the rotation speed is also known we are able to calculate the velocity of the ball $V = \omega \times R_{measre}$ where ω is the angular velocity converted from RPM of the motor and R_{measre} is the rotating radius. After measuring the empty domain, one magnetic bead is fixed on the rotating disk with glue at the radius 14.2 mm. The sampling frequency is set at 50 Hz. And the rotation speed of the disk is set at 10 rpm, which is equivalent to 1.0467 rad/s. Both trajectory and orientation histories obtained from the MPT measurements (Figure 20 b and c) show a typical circular motion and the diameter is almost identical with our calibration setting. A closer look at the Fig. 20 b) reveals that the period of the coordinate is 300 data sample (equivalent to 6 seconds), which matches well with the rotation disk speed. Figure 20 c) tells that the MPT method can also capture the spin or rotation of the particle successfully. The superposition image and the trajectory of the magnetic ball detected using image-based method is shown in figure 20 d). The errors in location of the particle will strongly contaminate the velocity result if using a simple finite difference method. Instead a more-robust numerical differentiation scheme that can also smooth the trajectory [83] is used here. The time derivatives measured from a weighted average of many

points is obtained by convolving the particle tracks with a Gaussian smoothing and differentiating filter.

The n-th-order derivative of the position is given by:

$$\frac{d^n}{dt^n} x(t) = \int_{-\infty}^{\infty} \tilde{x}(t - \tau) k_n(\tau) d\tau \quad (24)$$

where \tilde{x} denotes the noisy measured data or the trajectory of the tracer particle, $k_n(\tau)$ is the Gaussian filtering and differentiating kernel defined by

$$k_n(\tau) = \frac{d^n}{d\tau^n} \left[\frac{1}{\sqrt{\pi}\omega} \exp\left(-\frac{\tau^2}{\omega^2}\right) \right] \quad (25)$$

Integrating equation 24 by parts n times, the derivative in the kernel can be passed to the noisy position data; the surface term from each integration by parts vanishes since the Gaussian vanishes at infinity. However, the measurement cannot be infinitely long, then there will be a nonzero surface term due to finite integration interval. Therefore, the kernel needs to be truncated and renormalize before convolution with the particle trajectories. Taking the velocity as example, given by $n=1$. After truncation and renormalization, the kernel is given by:

$$k_v(\tau) = A_v \tau \exp\left(-\frac{\tau^2}{\omega^2}\right) + B_v \quad (26)$$

where the $\frac{1}{\sqrt{\pi}\omega}$ is absorbed into A_v . Let us restrict the integration domain to $[-T, T]$ instead from $[-\infty, \infty]$, and in principle, ω also needs to be set, for simplicity ω is fixed at $\omega = T/1.5$. To fix the constant A_v and B_v , we use the condition that the derivative of a constant should vanish and the derivative of t must be unity, combined these two conditions we have:

$$\int_{-T}^T k_v(\tau) d\tau = 0 \quad (27)$$

and

$$\int_{-T}^T (t - \tau) k_v(\tau) d\tau = 1 \quad (28)$$

By solving these two equations simultaneously, we are able to find A_v and B_v :

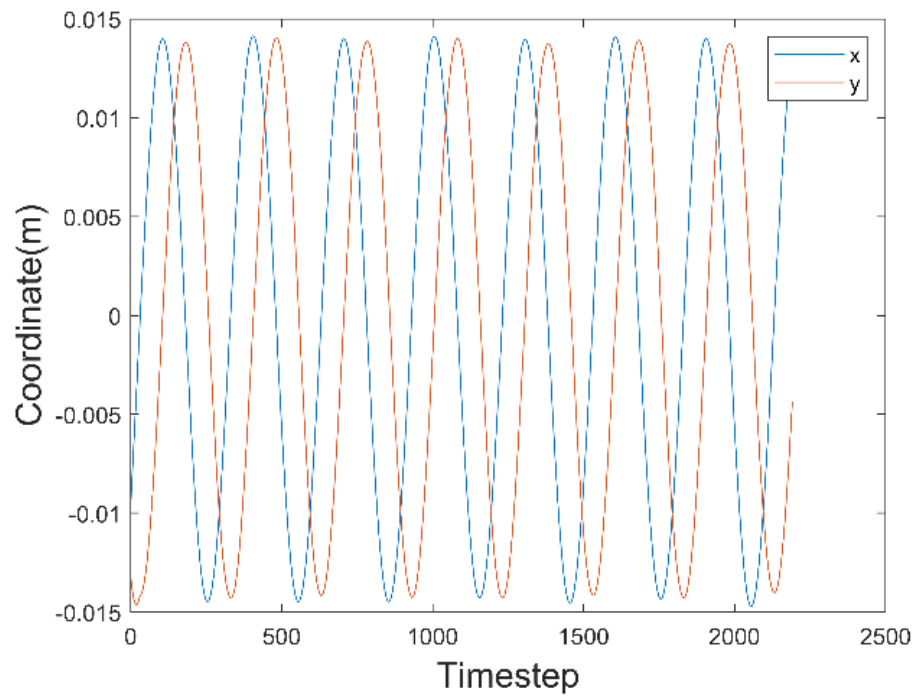
$$A_v = \left[\frac{1}{2} \omega^2 \left(\omega \sqrt{\pi} \operatorname{erf} \left(\frac{T}{\omega} \right) - 2T e^{-\frac{T^2}{\omega^2}} \right) \right]^{-1} \quad (29)$$

And

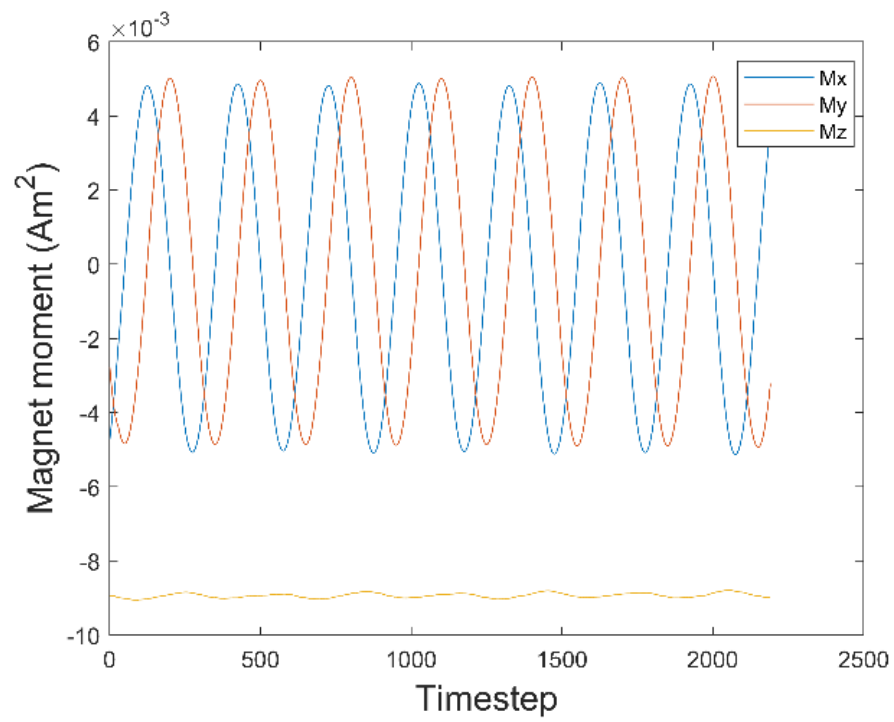
$$B_v = 0 \quad (30)$$

Applied this velocity kernel algorithms to the trajectory we obtained, the velocity profile is shown in Figure 20 e). The real velocity is 14.9 mm/s while the standard deviation of velocity is 0.65 mm/s, which is 4.36% of the velocity magnitude. Therefore, we can confidently apply the MPT method to our granular Couette experiment.

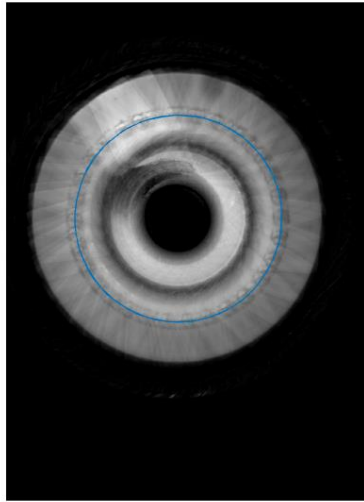




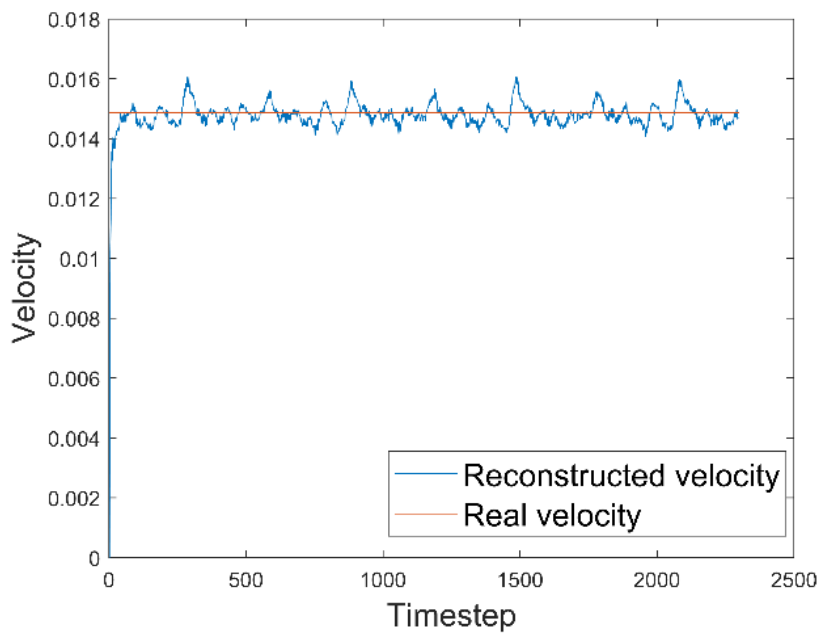
b)



c)



d)



e)

Figure 20 a) The reconstructed coordinates x and y of a fixed ball on the rotating disk with at 10 rpm. b) The magnet moment variation obtained from the MPT. c) The velocity profile of the ball fixed on the rotating disk.

5 Couette Cell Experiment

All experiments are conducted in an environment without any ferromagnetic material and the Couette cell used are the same one as shown in figure 25 a) but for different purpose of experiment the granular materials are different. The whole device is placed in a Helmholtz coil that balances the Earth magnetic field; in that way the earth magnet has no effect on the tracer particle. The radius of the cell container is $R = 50.5$ mm. The divided bottom has a rotating disk at the center. The radius of this rotating disk is $R_s = 31.7$ mm ($R_s/R = 0.63$). To ensure the bottom disk provides a consistent shear rate, one layer of granular material is glued on it. The disk is driven by a controlled DC motor at 10 rpm, and the shaft between the disk and motor is 65 cm long to ensure that the motor has little effect in the measurement domain. The magnetometers are Bartington M612 probes (3-axies) and are installed on two circular decks with an angular separation of 120 degrees. The gap between the decks is 37.5 mm. The magnetometers are located 67.5 mm away from the center of the cylinder. In the measurements, the sampling frequency is set to 50 Hz. We choose such a low frequency because the driving motion is slow. There are two types of granular material been used in our experiment for different purpose.

5.1 Small Diameter Spherical Copper Ball and Spherical Magnet with different shear rate

The first experiment we did in the Couette cell uses brass balls 260 grade 200 shown as figure 21 a). The diameter of the brass ball and the magnetic tracer ball are both $d = 3$ mm. The weight of the brass ball is 0.141 g, and the weight of tracer ball is 0.130 g. So, the tracer ball has nearly the same mechanical property as normal balls. There are totally 8-layer brass balls randomly packed in the cylindrical container. So, the filling height $H = 24$ mm and the $H/R_s = 0.7571$ which is considered as a large filling height. The diameter of the container is 101 mm, so the maximum

number of rings in radial direction is $51 \div 3 = 17$. Therefore, we divided the Couette cell into 8×18 spatial zones as shown in Figure 21b). The rotation speed of the bottom disk is set at 8,10,12,14,16,18 and 20 rpm which correspond to $\Omega = 0.84, 1.05, 1.26, 1.47, 1.68, 1.89, 2.09$ rad/s, respectively. To provide reproducible friction, the bottom disk surface has a layer of brass ball glued to it. The sampling frequency is set to 50 Hz in the experiments, and each experiment runs over 45 minutes. To avoid the initial condition effect, the tracer ball is pre-mixed in the brass balls and before starting recording data of each experiment, there is enough time for the Couette cell to mix the granular material.

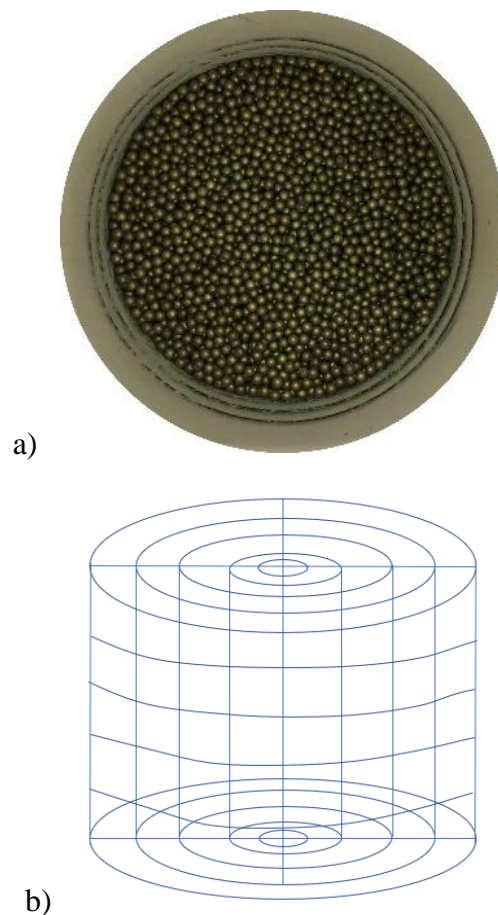


Figure 21 a) The $d=3\text{mm}$ copper ball granular flow b) The illustration of Couette cell is discretized into several zones in the radial and vertical directions.

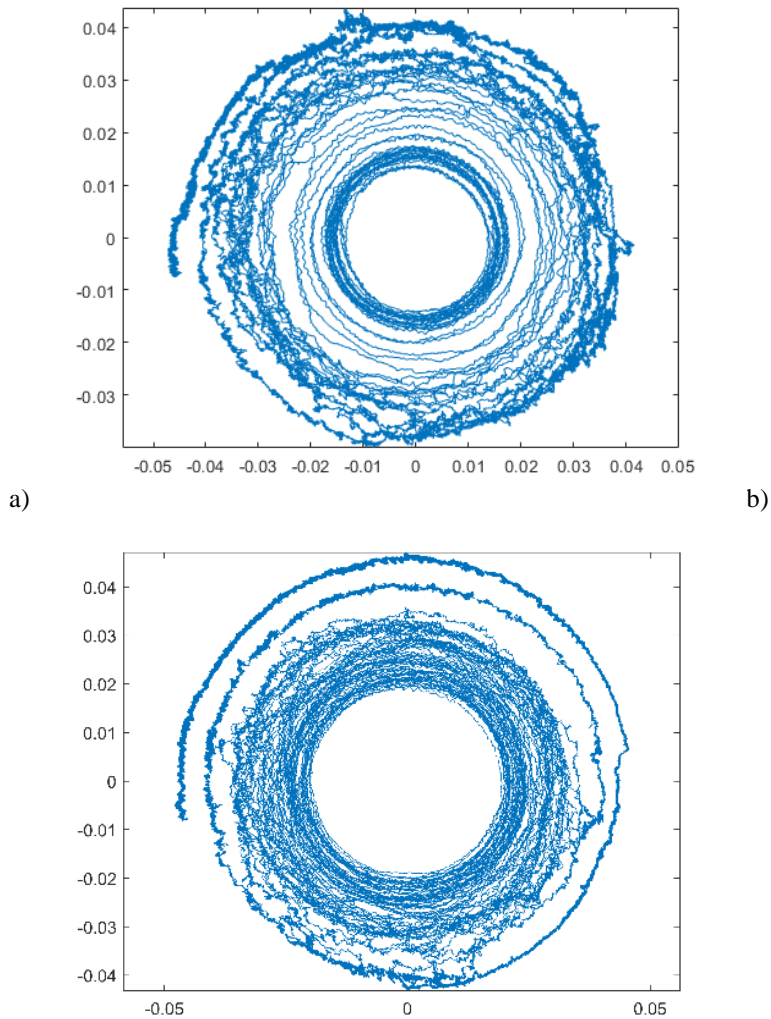
Using the EKF method mentioned above, we can reconstruct the trajectory of the tracer particle. The reconstructed trajectories at different rpms show a clear layered distribution (Figure 22). It is obvious that the paths are not evenly distributed in the radial direction. Take a scrutiny of these trajectories, the tracer ball is more likely to appear on the inner and outer circles while passes through the middle section faster. This is easy to understand: since the shear bands are the regions of large velocity gradient, the tracer ball paths in these areas are more stimulated than others, therefore the staying time among these areas will be shorter. The numbers of recorded data are different for these figures, and there is no clear indication that different rpms influence the staying time of the tracer ball at each zone. However, by collecting the number of times tracer particle appears in each zone, we are able to figure out at which area the tracer particle stays the longest and shortest.

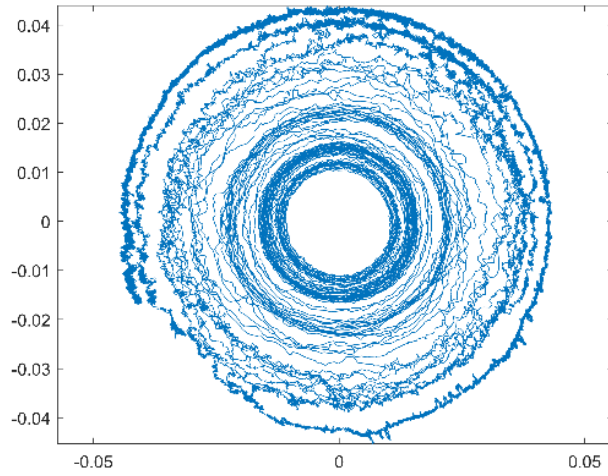
Contours of the appearance density distribution are shown in Figure 23. Since the actual size of the zone is linearly increasing with the radius r , we normalize the appearance density of each zone by dividing $2\pi r$; therefore, the density profile only depends on variables r and H . Although Figure 23 a) to g) vary significantly, the highest density always appears at the zones whose radius is around 2 and followed by the zones whose radius is 14-16. Here, to be clear, the density means the number of times the tracer ball appears. In that sense, the tracer ball is more likely staying close to the inner ring and the outer boundary. This seems to agree with the earlier research result that velocity profile is described by an error function as eq. 31 [61], [84]–[86].

$$\omega(r) = 1/2 + 1/2 \operatorname{erf}((r - R_c)/W) \quad (31)$$

where R_c and W denote the center of the shear band and the widths of the shear band. We define the $w(r)$ as the dimensionless ratio of the average angular velocity and driving rate Ω . The density

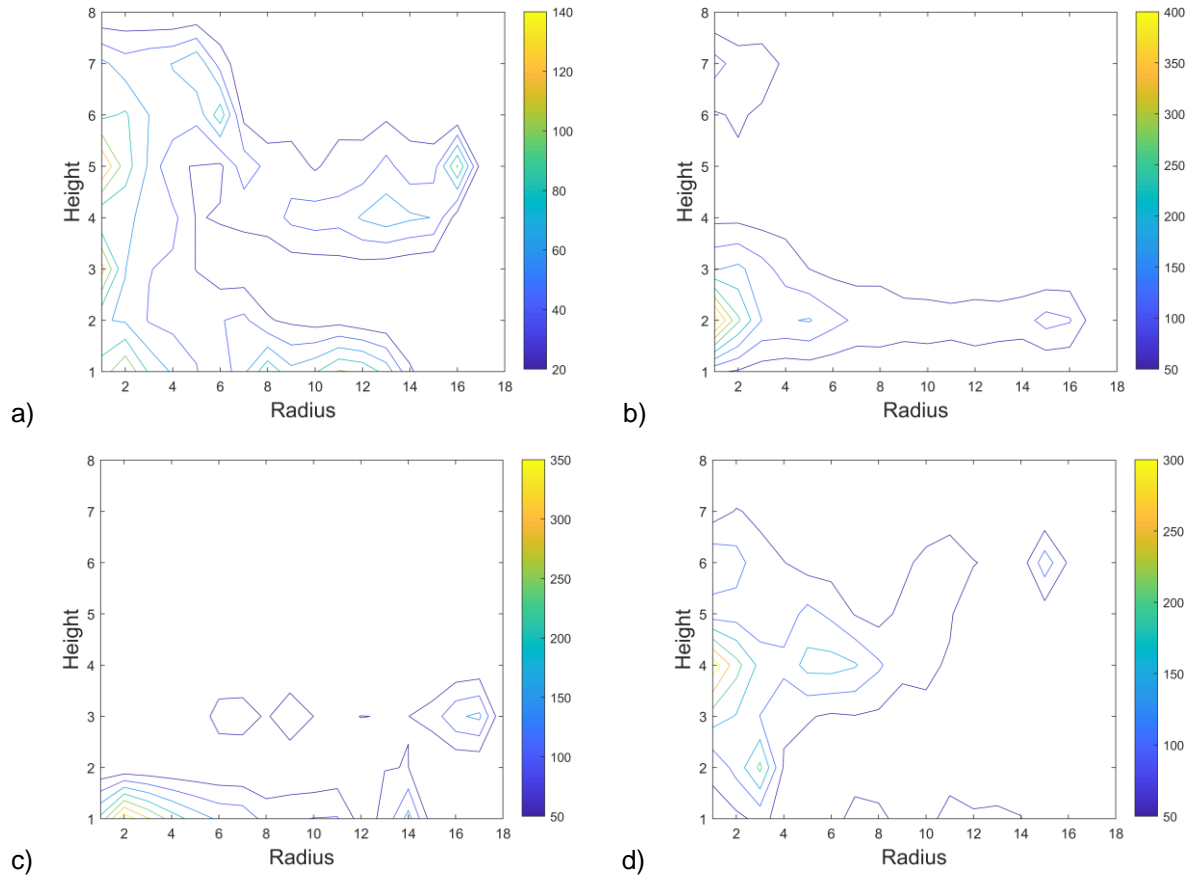
is the lowest at the center of shear band where the derivative of the velocity is the highest. Because the shear bands are the regions of large velocity gradients, the granular materials collide and exchange their energy, which stimulates each individual particle, and it is less likely to be stable at this region. Therefore, the tracer particle has a lower chance to be in the shear band zone. Though, Sakaie [87] points out that eq. 31 works well for low filling heights $H/R_s \leq 0.55$ while with the increasing of filling height the flow profile at the free surface deviate from eq. 31. Since the device is also different and our granular material is 10 times bigger than that used in Fenistein et al. [86], the velocity profile of the tracer particle is not exactly described as eq. 31.





c)

Figure 22. Three sample trajectories. a), b), and c) are corresponding to 8,10 and 20 rpm respectively.



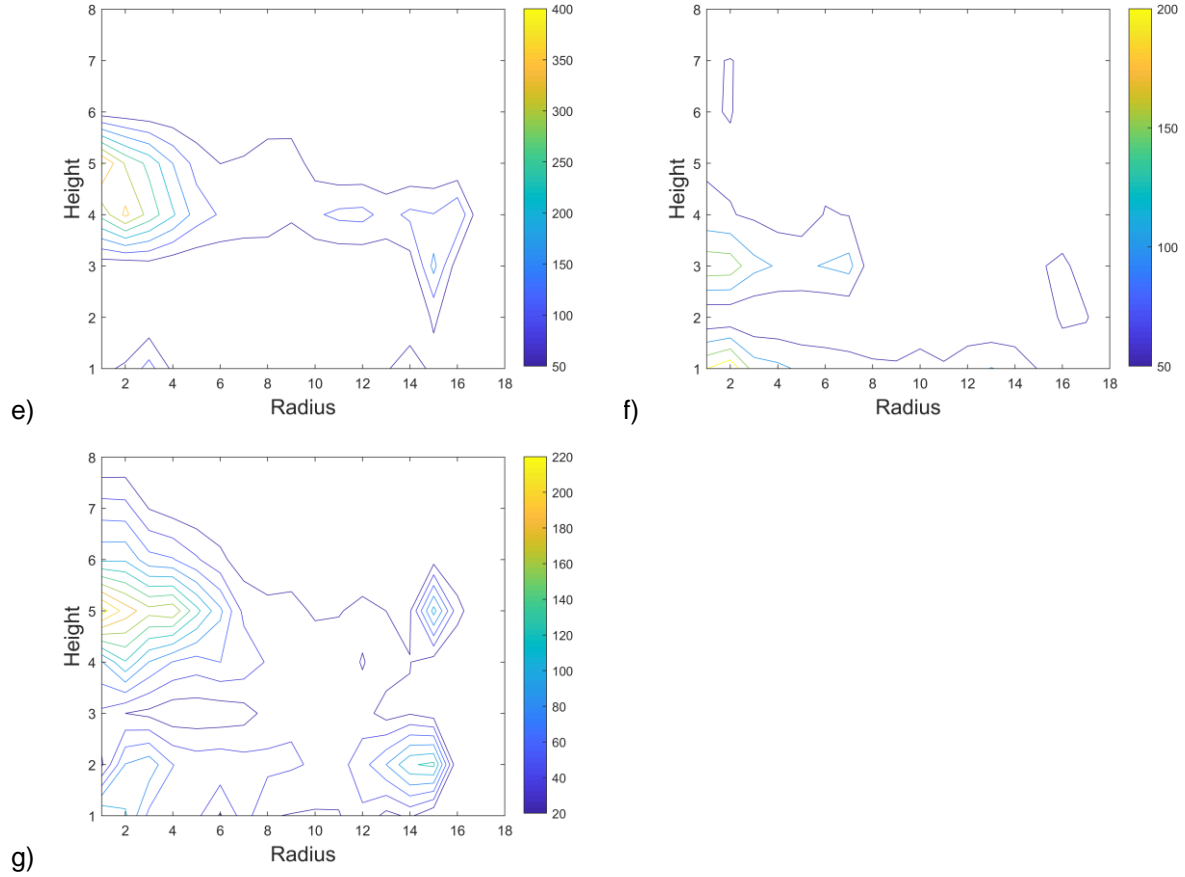


Figure 23. Density contours of the tracer particle appearing at each zone for a), b), c), d), e), f), g) are 8,10,12,14,16,18,20 rpm respectively.

It is able to get the velocity profile in this modified Couette flow system at different rpms by collecting the velocity of particle appearing in the same zone and averaging them. Since the $R_s \approx 10d$, $R_c \approx 13d$ and the widths of the shear bands $W \approx 7d$, input these into equation 31.

$\omega(r) = 1/2 + (\frac{1}{2} \text{erf}(\frac{r-13d}{7d}))$. The comparison of this function and the measured $\omega(r)$ against radius at different rpms is shown in Figure 24. We notice that the velocity profile matches better when the radius is beyond $8d$, but the measured results have large fluctuations and deviate from the error function at $r < 6d$. One of the reasons is the limited number of samples at the center (the absolute number of samples at a small radius such as $1d-3d$ is very small). Therefore, it may not

have sufficient data to produce a converged velocity profile in these areas. Extending the experiment duration may be necessary in the future works. Another possible reason could be the little step between the rotating disk and Couette cell bottom changes the velocity profile at where closer to the disk.

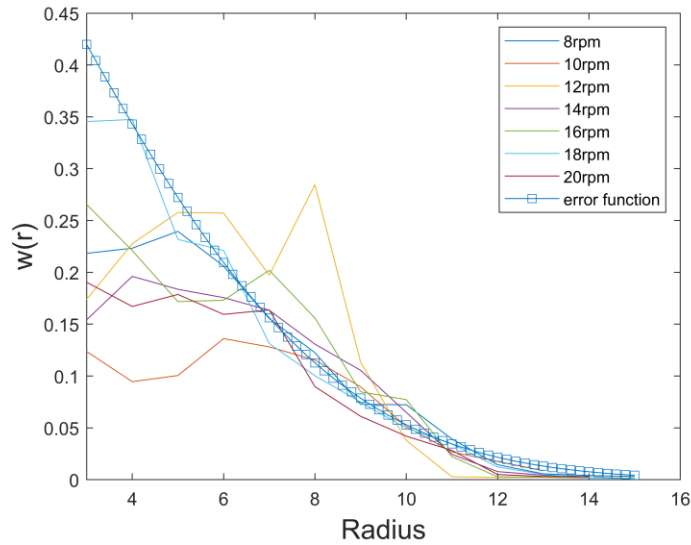


Figure 24. Velocity profile showing the ratio of average angular velocity, with different rpm and compared with the error function $\omega(r) = 1/2 + (1/2 \text{ erf} ((r - R_c)/W))$, here $R_c = 13d$ and $W = 7d$.

By collecting and averaging the v_θ and v_r at the same height, we find that the v_r component is relatively very small when the tracer ball is under free surface, namely height $< 8d$. Our v_θ result does not show any common characteristics regarding different rpms as shown in Figure 28. We also notice that the amplitude of v_r is generally smaller than the v_θ which is expectable, because the energy source of this Couette cell system is providing an azimuthal motion. The result and conclusion will be discussed in chapter 5.3.

5.2 Small copper ball result

We have demonstrated the usage of a new technique, MPT, in the study of dense granular flow. This single-particle level tracking technique is able to measure the particle 3D trajectory in an optically not accessible environment. The position accuracy of MPT is comparable with optical based methods and MRI, but its sampling frequency could be much higher than MRI. We have found our experiment data matches well with previous research (i.e., the mean angular speed is an error function) when the radius is larger than $8d$. However, the result deviates and fluctuates a lot when the radius is less than $6d$. This is partially due to the limited number of samples in this preliminary study. Note that the Lagrangian method requires a very long time series data to produce a converged velocity profile. We also notice that the granular material used in this project are metal balls which may cause magnetic field change when they move and cut the magnetic field lines. But since the diameter of the ball is very small and it is a very weak magnet strength the effect is negligible.

In the future, we will carry out much longer measurements to obtain the Lagrangian trajectories with sufficient samples and substitute the granular materials with plastics. This will allow us to calculate the velocity fluctuations in the shear layer (i.e., the granular temperature distribution) and the diffusion coefficient with less disturbance in the system. Since the MPT method provides the orientation of the tracer particle, we will study the alignment between the eigenvectors of the mean shear stresses and the orientation of particles. Moreover, the mixing of poly-dispersed particles can also be investigated using our technology. The mixing problem is of great interest to industry.

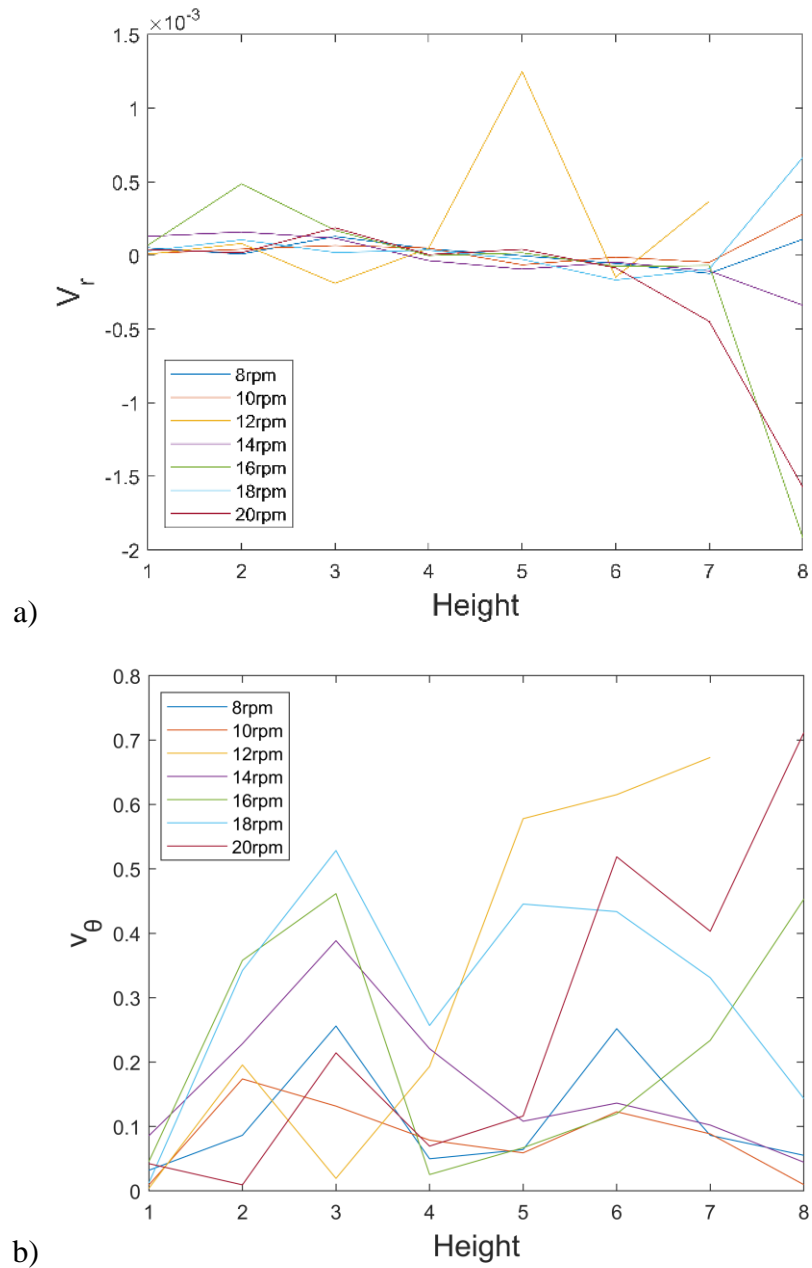
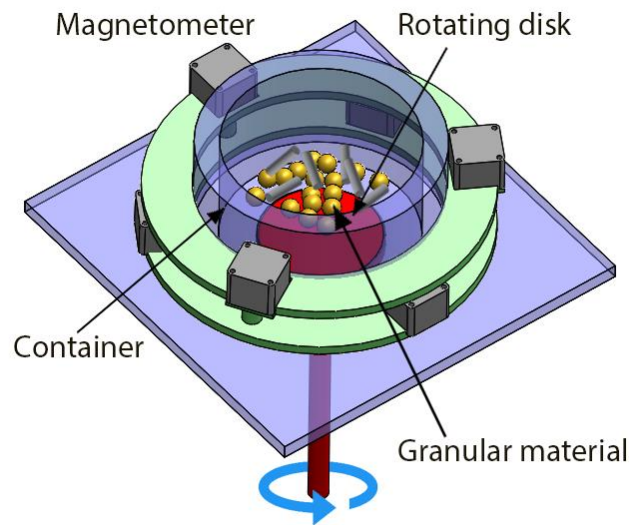


Figure 25 a) and b) are v_r and v_θ against height at different rpms respectively.

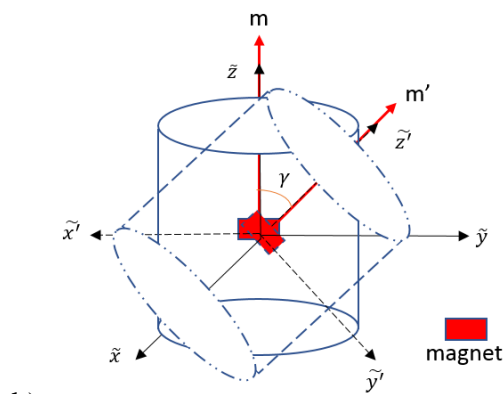
5.3 Cylindrical plastic tracer particle with a constant shear rate

There are many researchers have investigated the behavior of non-spherical granular material with different aspect ratio in a shear flow. In this experiment a cylindrical particle with an aspect

ratio of 1 serves as the tracer particle and interacting with other spherical granular materials. The granular material used here is made of Delrin® Acetal Resin Balls (FIG. 25 c). These balls resist expanding when exposed to heat and moisture and have a naturally slippery, wear-resistant surface. The diameter of the sphere is 12.5 mm. The filling height $H = 50$ mm. The cylinder diameter is also 12.5mm and its aspect ratio is 1. A small neodymium coaxial cylindrical magnet is inserted into the core to label this cylinder (FIG 25b). The trajectory of this sample particle can be reconstructed using the magnet field and our MPT algorithm, as described in detail below. The measurement takes more than 1.5 hours, and 300,000 data points are recorded.



a)



b)

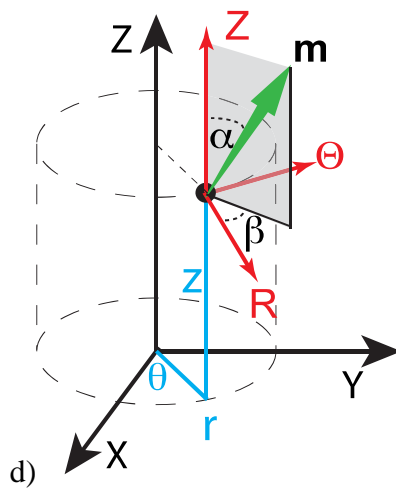
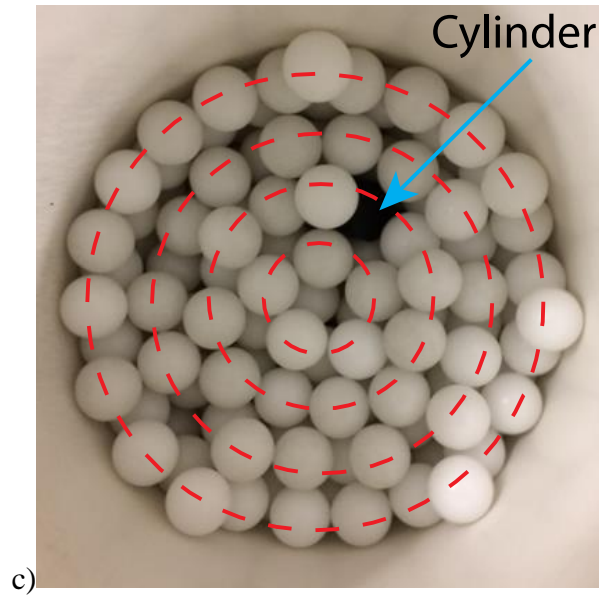


Figure 26 a) A schematic of the experimental setup. b) The diagram of the tracer particle, one coaxial cylindrical magnet is inserted in it. The magnet moment is always aligning with the local \tilde{z} -axis. So, the rotation measured is either around \tilde{x} of \tilde{y} - axis. c) The top view of the surface in the Couette cell. d) Definition of the cylindrical coordinate ($r \theta z$) and locate frame ($R \Theta Z$). Here m indicates the magnetic moment.

The MPT method provides the Lagrangian trajectory of a sample particle. A section of trajectory is shown in FIG 26. The red arrow indicates the particle orientation (the magnetic

moment) at each location. It is more convenient to use a cylindrical coordinate to describe our result (see FIG 25 for coordinate definition). The trajectory is $\mathbf{r} = [r(t), \theta(t), z(t)]^T$ and orientation $\mathbf{m} = [mr(t), m\theta(t), mz(t)]^T$. The trajectory can be used to calculate, for example, the velocity $\dot{\mathbf{r}} = [V_r, V_\theta, V_z]$. However, the measurement uncertainty is usually magnified when calculating derivatives using finite difference. Taking V_z as an example, the velocity spectrum is flat up to 5 Hz (FIG 27). Note that the sampling frequency is 50 Hz. Beyond 5 Hz, the spectrum fluctuates considerably, indicating the existence of high frequency noise. This noise can be mitigated using a low-pass filter,[88][89] such as a Gaussian filter $G(\eta) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\eta^2}{2\sigma^2})$, where σ is the window size. Given a sequence of measured positions Z^* , which contains uncertainty, the filtered position is

$$Z(t) = \int Z^*(t - \eta)G(\eta)d\eta \quad (32)$$

and hence, the velocity is

$$V_z = \frac{dZ}{dt} = \int \frac{dZ^*}{dt} Gd\eta = \int Z^* \frac{dG}{d\eta} d\eta \quad (33)$$

In the last step, the integration by part is used. This method has been widely used in particle tracking.[88] To maintain the flat part of the spectrum, we choose a window size $\sigma = 0.2$ s, which corresponds to 5 Hz and contains 11 data points in the sequence. Therefore, the resulting temporal resolution is in the order of 0.2s. The spectrum of the filtered velocity is shown in FIG 27 b). The high frequency noise is significantly reduced. The same filter is also applied to the director \mathbf{m} .

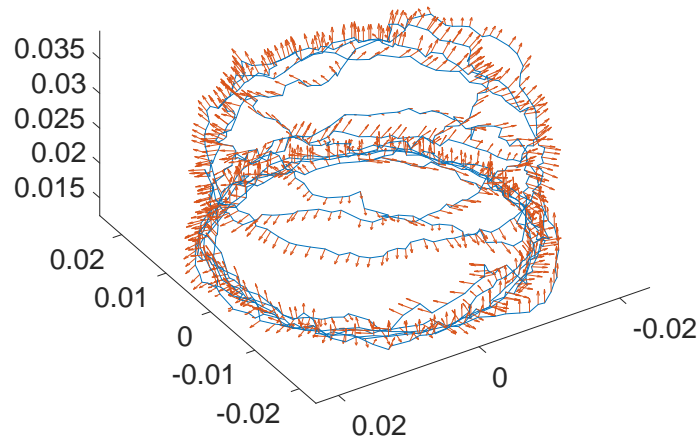


Figure 27. A sample section of trajectory. The red arrows are the direction of the cylinder particle. The unit of axis is meter.

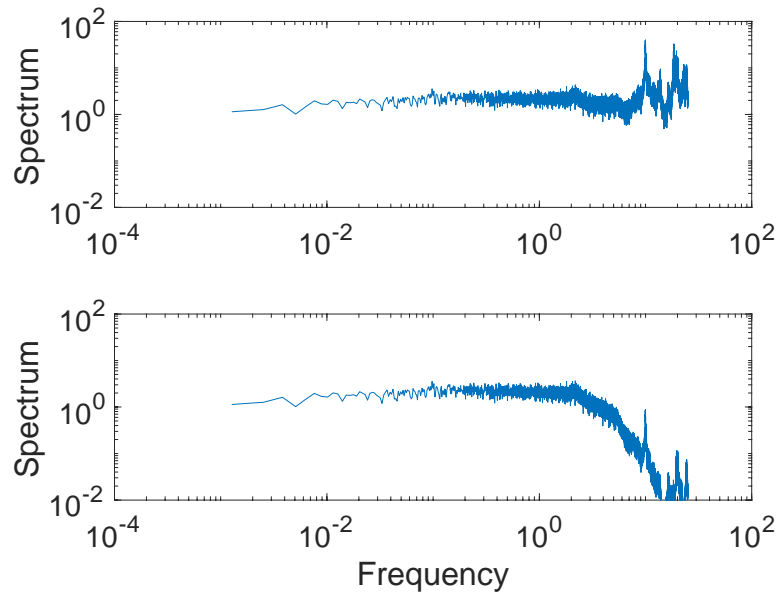
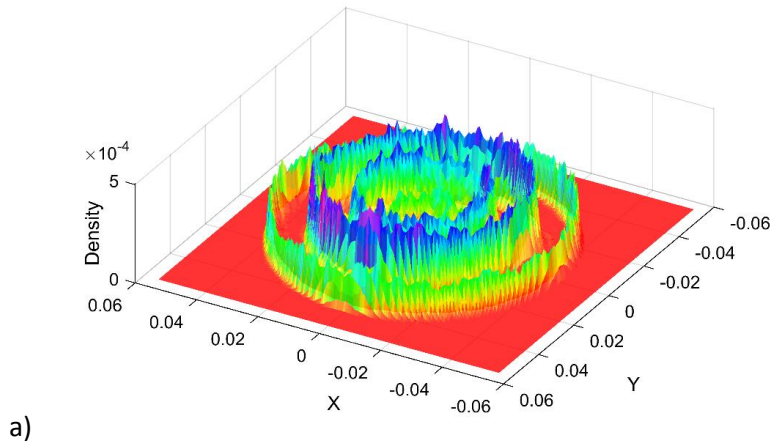


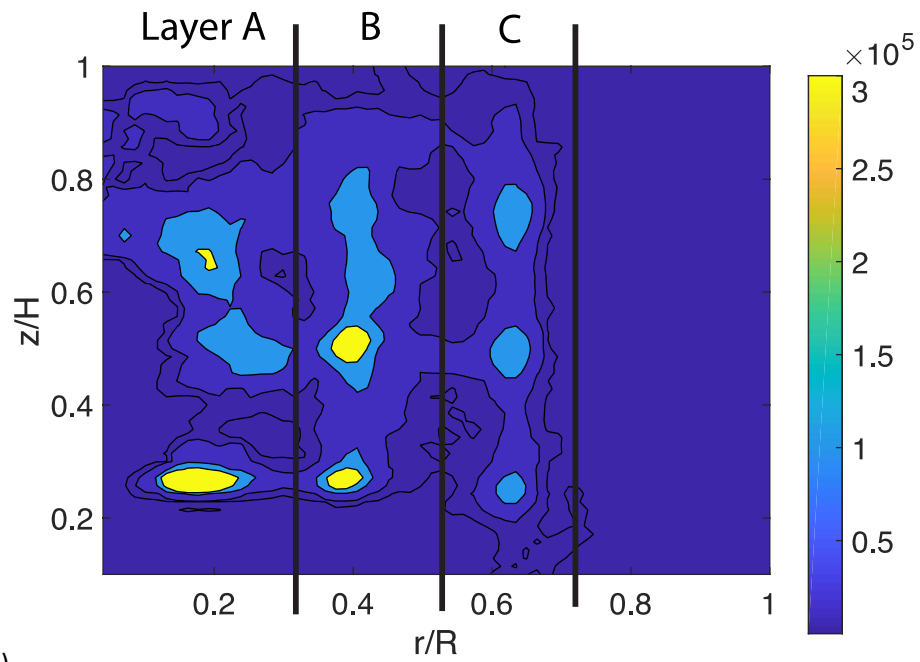
Figure 28. The power spectrum of V_z . Without filtering, the position uncertainty results in high-frequency noise. The Gaussian filter effectively reduces the noise. The frequency unit is Hz.

5.4 Cylindrical plastic ball result

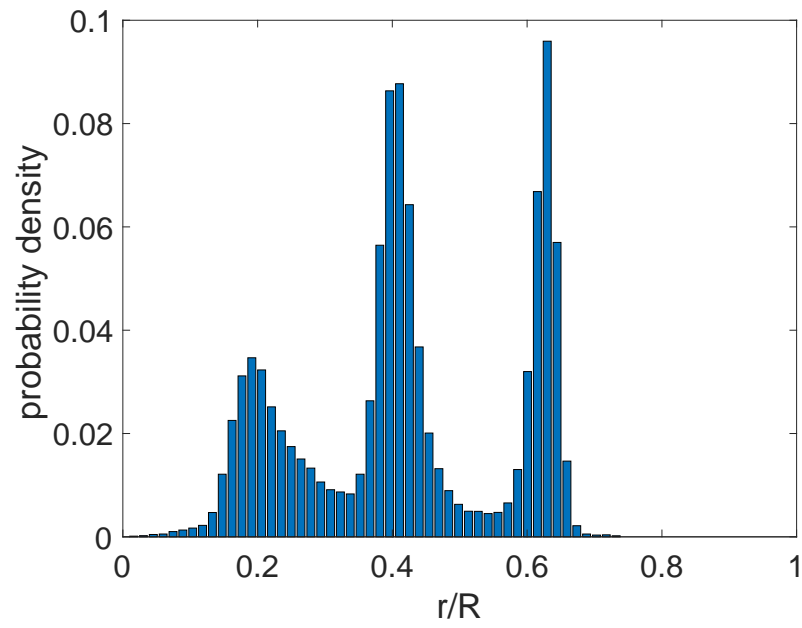
5.4.1 Position statistics

Observed from the surface (FIG. 25b), the tracer particle travels in a layered structure. FIG. 29 a) is the histogram of how long the tracer particles remain in the x - y position. It indicates that the bulk flow also possesses a layered structure. Assume the position probability density distribution of the cylinder is $\rho(r, \theta, z)$, the r - z marginal distribution is $\rho(r, z) = \int \rho(r, \theta, z) d\theta$. Note that the probability distribution is normalized as $\int \rho(r, \theta, z) r dr d\theta dz = 1$. Assuming the system is ergodic, we can obtain the density distribution based on the trajectory. Practically, the density can be calculated using kernel-based density estimation methods[90], [91] Figure 29 shows the $\rho(r, z)$ and 1D marginal distributions $\rho(r) = \int \rho(r, z) dz$ and $\rho(z) = \int \rho(r, z) r dr$. The number density concentrates at a few islands on the r - z plane, indicating that the cylinder travels in a lattice-like bulk flow. The segregation in the R direction is marked using black lines on FIG. 29b. For the convenience of discussion, we define three layers A ($0 < r/R < 0.34$), B ($0.34 \leq r/R < 0.54$), and C ($0.54 \leq r/R < 0.74$) in the R direction. In our measurement, the cylinder has rarely reached the outer layer, i.e., the $r/R > 0.74$ region.





b)



c)

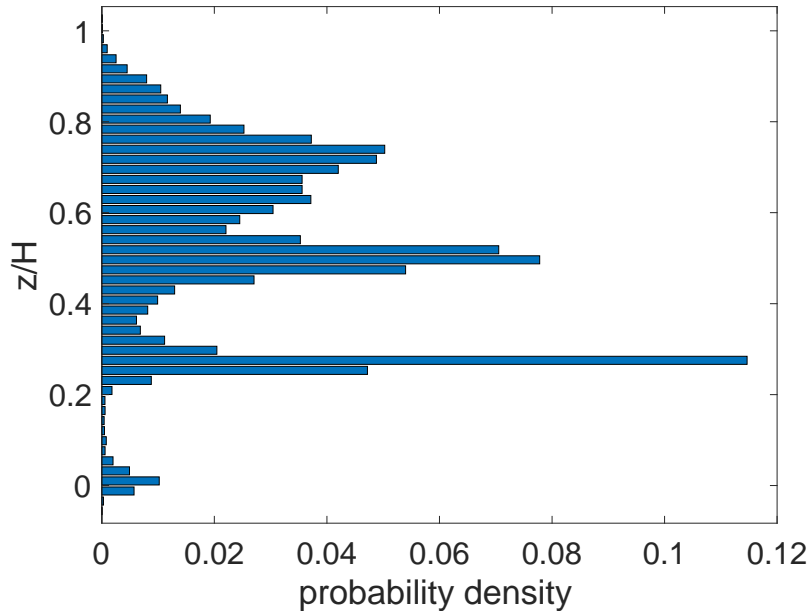


Figure 29. a) The number density distribution along radius direction, this result compressed all data in the z -direction into x - y plane. b) Number density distribution $\rho(r, z)$ obtained from the trajectory of the sample cylinder. The bulk flow shows a layered structure. c) The marginal probability distribution $\rho(r)$. d) The distribution $\rho(z)$ drawn in the vertical direction. The layers in R direction are labeled as Layer A ($0 < r/R < 0.34$), B ($0.34 \leq r/R < 0.54$), and C ($0.54 \leq r/R < 0.74$).

The layered structure leaves its signature in the velocity field. The average velocity in Θ direction $\langle V_\theta \rangle$ is shown in FIG. 30 a). It remains relatively uniform in the Z direction, but from one layer to another, $\langle V_\theta \rangle$ jumps a step. FIG. 30 b) shows the average magnitude $\langle |V_r| \rangle$ (the mean value $\langle V_r \rangle$ is nearly 0). The $\langle |V_r| \rangle$ distribution shows a grid-like structure, and the value in the layer gap is higher. In addition, $\langle |V_r| \rangle$ is also higher near the top surface, as particles there have less constraints. These velocity and numerical density distributions indicate that the sample particle has preferred radial locations (i.e., staying in the layers) and the particle transition between layers is a jump with a higher $|V_r|$.

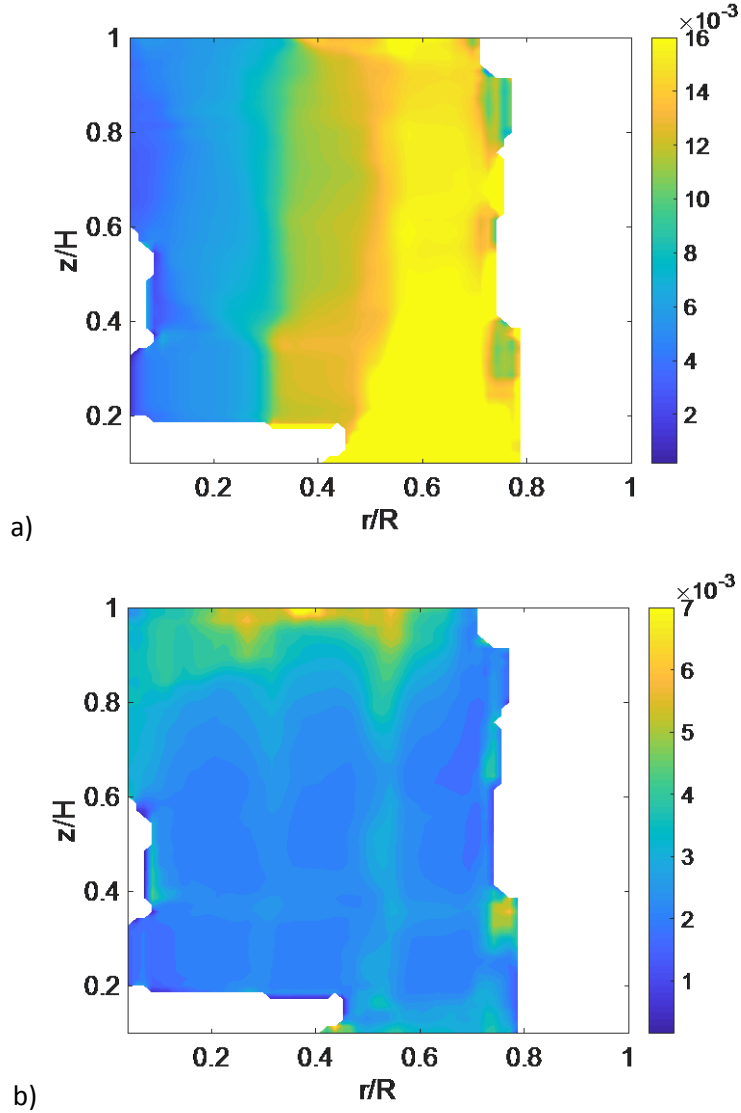
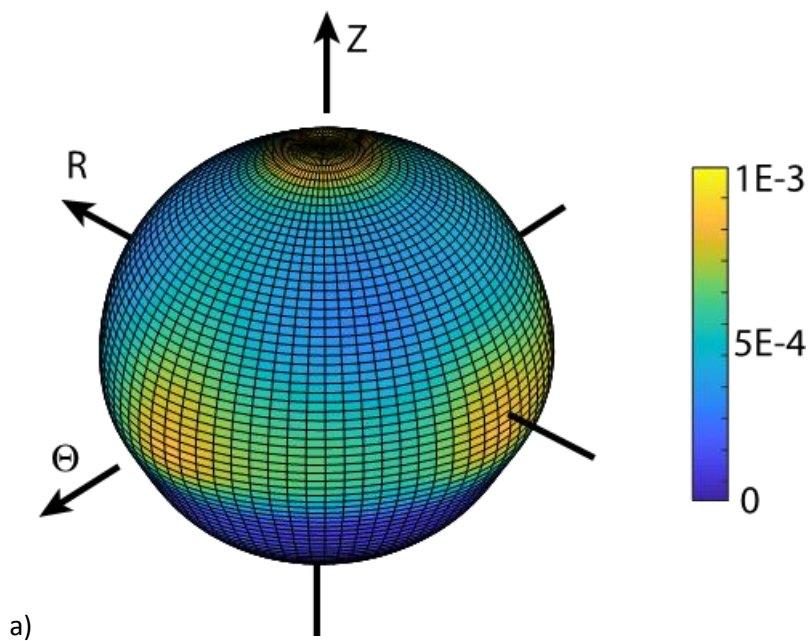


Figure 30 a) The average velocity $\langle V_\theta \rangle$ and b) the average velocity magnitude $\langle |V_r| \rangle$.

5.4.2 Orientation of the cylindrical particle

Usually, only high aspect ratio particles have a significant alignment in a shear flow, but our results demonstrate that a cylinder with an aspect ratio 1 clearly has preferred alignment. The spherical probability distribution $p(\mathbf{m}) = p(\alpha, \beta)$ illustrates the orientation preference (FIG 30a). Here, α and β are the solid angles defined in Figure 1c, and the probability is normalized in the following way: $\int p(\alpha, \beta) \sin(\alpha) d\alpha d\beta = 1$. The cylindrical particle has three preferred

directions, $\pm R$, $\pm\Theta$, and $\pm Z$. This alignment preference can also be observed in the 3D trajectory plot (FIG 25), in which the red director arrow usually points towards the $\pm R$, $\pm\Theta$, and $\pm Z$ directions. Because of the small aspect ratio, the alignment of the sample particle cannot be directly caused by shear. In addition, such alignment cannot occur to an isotropic spherical particle. Therefore, the alignment is largely due to the anisotropic shape and contacts in a layered flow. The meridional section of the sample cylinder (i.e., the axial cross-section) is a square. If the particle is not aligned (FIG 30b), the intrusion encounters other particles, which provide a torque to re-align the cylinder. The most probable orientation is, therefore, to have the flat surfaces tangential to the layers, i.e., the particle may align with the R Θ or Z axis.



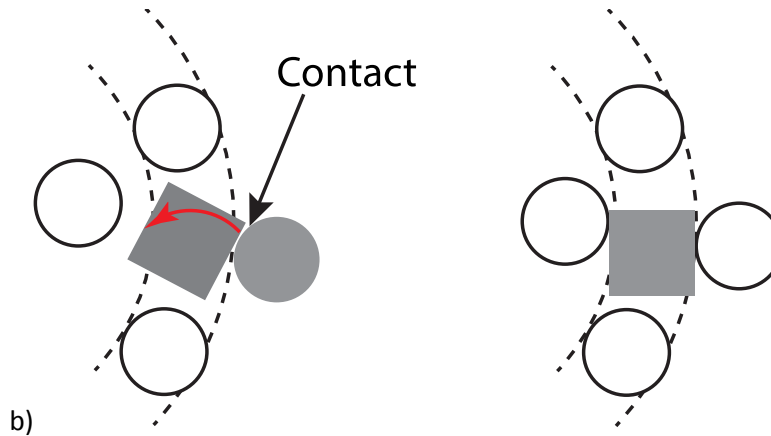


Figure 31 a) The orientation distribution of the sample particle. The particle prefers to align in the R, Z or Θ direction. The drawing utilizes an open-access MATLAB code. b) The contact between a cylinder and particles from another layer helps align the cylinder to be in a preferred direction.

5.4.3 A Symbolic Method to Study Flipping and Jumping of the Tracer Particle

The above results suggest that the cylindrical particle has preferred positions (Layer A, B, and C) and orientations (directions R Θ Z). These states are discrete, and the particle jumps or flips between different layers or directions. In this section, we will study these transitions. The particle has infinite possible ways to jump from Layer A to B or to flip from R to Z direction. We are interested in the properties such as the intermittency of transition, whereas the exact trajectory of a specific transition process is not our focus at the moment. Consequently, to analyze the intermittency and transition of the particle, we will use the symbolic dynamics approach, which is a powerful tool for describing discrete statuses and transitions. The first step is to symbolize the position or orientation of a particle. Taking the orientation as an example, if the particle director is near the Z-axis (the angle between \mathbf{m} and \mathbf{Z} is less than $\pi/6$), we label the particle status at this moment as S_Z . Using other angle criteria for symbolization provides similar results. Consequently, we define S_R and S_Θ . If the particle is not in states of S_Z S_R or S_Θ , it is

considered to be flipping, which is a transition process and labeled as T. Subsequently, the orientation history along the Lagrangian trajectory can be translated into a sequence of symbols, e.g., $O = \{S_Z S_Z \dots S_Z S_Z T T \dots T S_R S_R S_R \dots S_R \dots\}$.

Examining the entire sequence, we find that there are two kinds of “T” sections (FIG 31). One is a normal transition, i.e., a section between two different kinds of symbols, e.g., $S_Z \rightarrow T \rightarrow S_R$; the other type connects the same symbol, e.g., $S_Z \rightarrow T \rightarrow S_Z$. The latter one is not a real transition; it corresponds to a large fluctuation away from the preferred direction. Therefore, the second type of “T” sections in the sequence must be filtered. Specifically, if a section of T in the sequence is sandwiched between the same symbol S_α ($\alpha = R, \Theta$ or Z), this section is replaced by S_α (FIG 7). Thereafter, we obtain the filtered sequence \hat{O} .

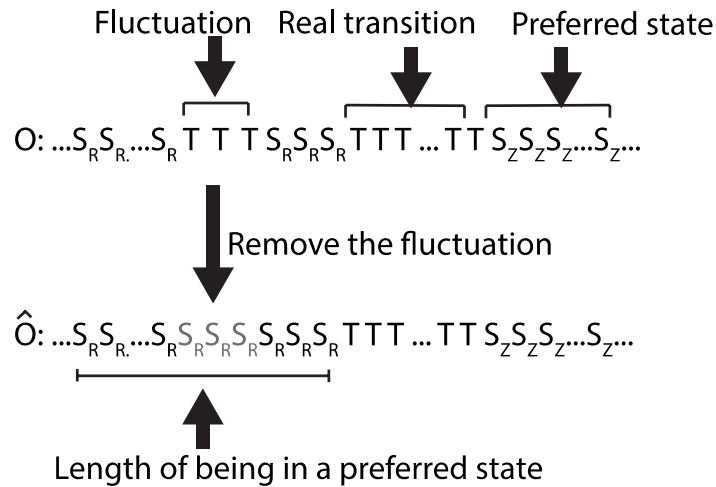
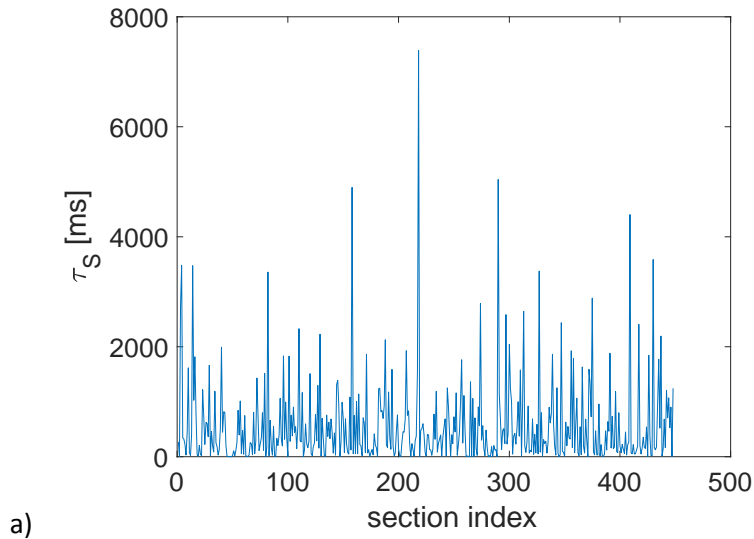


Figure 32 The status of a particle described using symbols. If a T section is in between two sections with the same symbol, this section is filtered because it is not a real transition (not flipping between different directions).

The length of a section in the filtered sequence \hat{O} indicates the duration τ of being in a preferred state (or in a transition process), as illustrated in FIG 31. Examination of \hat{O} shows that flipping is highly intermittent, and hence, the value of τ varies randomly (FIG 33a). We distinguish the duration of being in a preferred direction S or in transition T. The histogram of τ_S (for preferred states) and τ_T (for transition) are shown in FIG 33b. These histograms are normalized by the total number of S_α or T sections. The curve decrease considerably as τ increases. In other words, it is possible to stay in a preferred state for a long time, but this probability becomes infinitely small as the duration increases. The curves in FIG 33b cannot be represented using a power law. For example, the blue τ_S curve has a kink as indicated by the black dot line. The histogram of available data at a large τ suggests that the probability decreases fast enough, so the mean values of τ exist: $\langle \tau_S \rangle = 560$ ms and $\langle \tau_T \rangle = 140$ ms. In other words, the particles spend a longer time in a preferred orientation than in the flipping process. This fact is also manifested by the observation that the τ_S curve is located to the right of τ_T in FIG 33b.



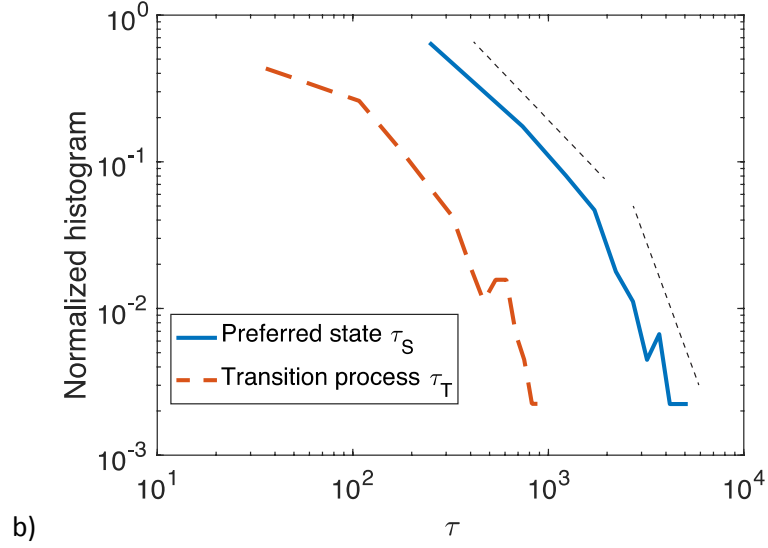


Figure 33. The durance of being in a preferred orientation. It is highly intermittent. b) Histograms of the durance of a preferred orientation or transition process.

Although the time of being in one state is highly random, the flipping may follow a determined order. For example, the particle may always flip from R to Θ and back to R direction, which reflects that the particle is rotating on the R- Θ plane. To evaluate how random the order of flipping is, we here examine the patterns of five successive sections (containing 3 preferred states and 2 transitions) in the \hat{O} sequence, e.g., a $S_R \rightarrow S_\Theta \rightarrow S_R$ section. There are 12 possible patterns, as listed in FIG 34. If the flipping is a completely random process, all patterns have an equal probability $1/12$. The associated Shannon entropy, as a measure of randomness, is $\Omega_0 = \log_2 12$. The experimentally obtained probability distribution of these patterns is non-uniform (FIG 34), and the entropy is $\Omega = 3.3 = 0.93 \Omega_0$. Hence, the above results show that the flow system, to some extent, determines where the cylinder particle flips, but does not determine when the flip occurs. The most probable order of flipping contains two groups: 1) $S_\Theta \rightarrow S_Z \rightarrow S_\Theta$ and $S_Z \rightarrow S_\Theta \rightarrow S_Z$, and 2) $S_\Theta \rightarrow S_R \rightarrow S_\Theta$ and $S_R \rightarrow S_\Theta \rightarrow S_R$. This reveals that the particle tends to rotate in the Θ -Z or R- Θ planes.

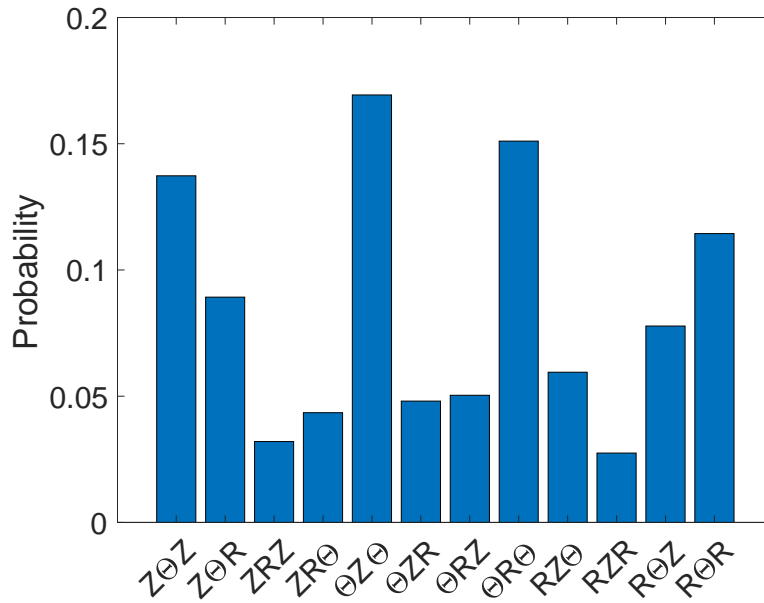


Figure 34 The probability distribution of all flipping orders in a $S_\alpha \rightarrow S_\beta \rightarrow S_\gamma$ sequence.

A similar symbolic process is used to analyze the jumping between the layers. Within Layer A, we symbolized the region of $r/R < 0.26$ as S_A . In Layer B, the region of $0.37 < r/R < 0.48$ is labeled S_B . In Layer C, S_C marks $r/R > 0.54$. The states in between are symbolized as T, i.e., regarded as in a transition or jumping process. Through the same filter process as orientation (FIG 32), a position symbol sequence (e.g., $\hat{P} = \dots S_B S_B T T T S_C S_C \dots$) can be obtained. Similar to the orientation results, the duration of being in a preferred position (roughly speaking, the center of each layer) is much larger than that of the transition process (FIG 35). This result indicates the jumping between layers is a rapid process and underscores the observation that $\langle |V_r| \rangle$ is larger in the gap between the layers (FIG 30b).

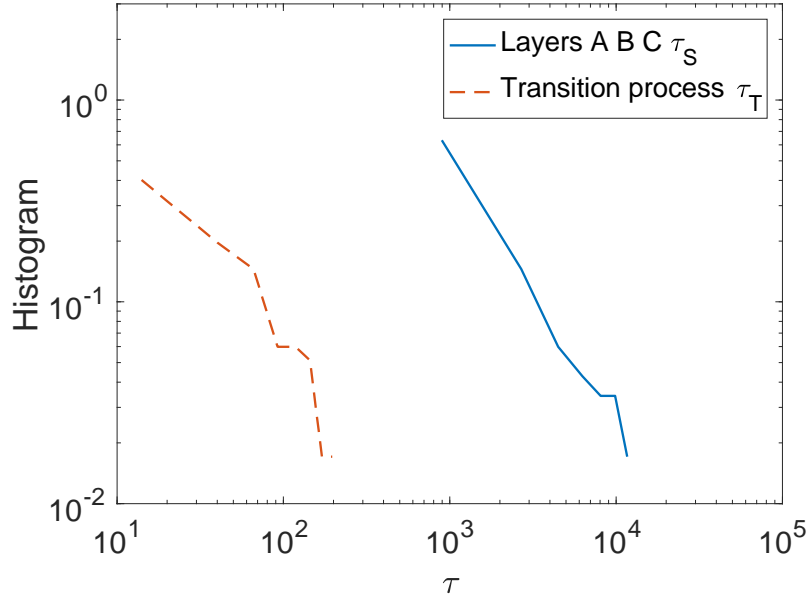
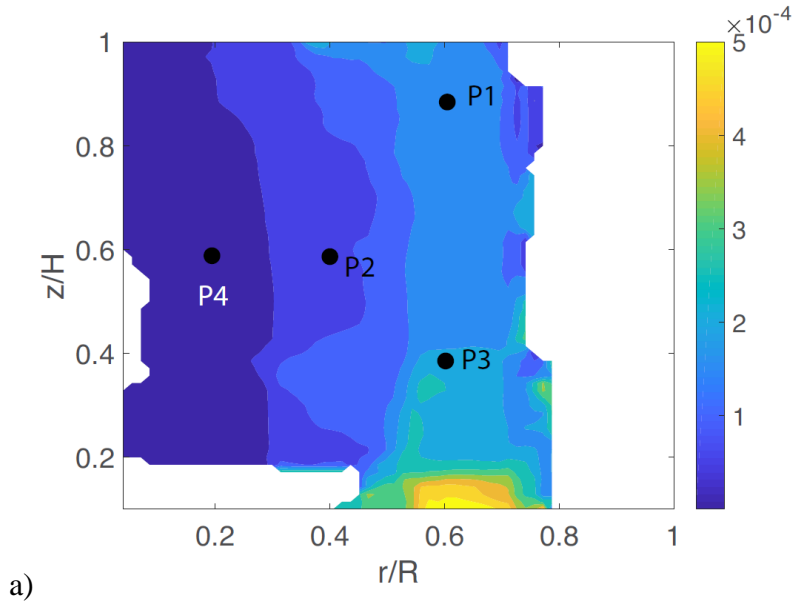


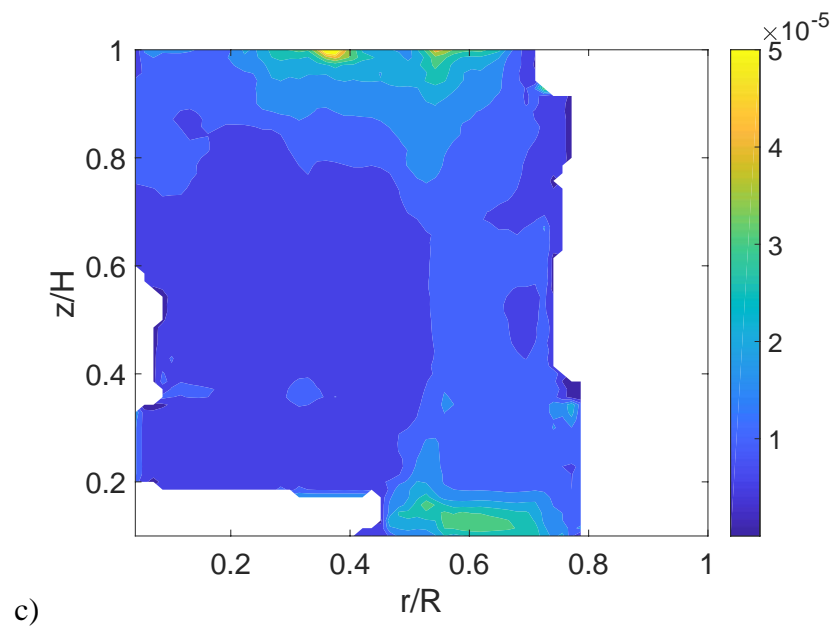
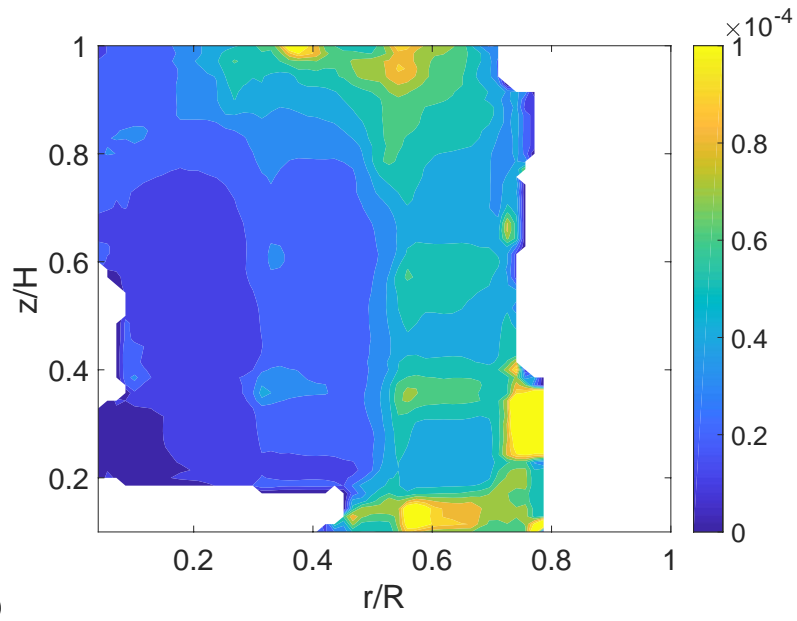
Figure 35 The histograms of durance of being in a preferred layer and in a jumping process.

5.4.4 Relationship between translation and rotation

The above Section IV.C has analyzed the translation and rotation as independent modes. In this section, we will compare the translational and rotational kinetic energy distributions and investigate their relationship. The translational kinetic energy is $E_T = \langle \frac{1}{2} \mu V^2 \rangle$ (FIG 36a), where μ is the mass of the particle and $V^2 = V_r^2 + V_\theta^2 + V_z^2$ can be calculated from the trajectory (see section III). In addition, the velocity fluctuation is critical to granular motion. Therefore, we need to pay attention to the energy of fluctuation $E_T' = \langle \frac{1}{2} \mu v^2 \rangle$ (FIG 11b), where $\mathbf{v} = \mathbf{V} - \langle \mathbf{V} \rangle$ is the fluctuation. Note that the granular temperature $\theta_T = \frac{1}{N} \langle v^2 \rangle$ is widely used in the kinetic theory where N denotes the spatial dimension. But here, since the motion of tracer particle occurs mostly in the x-y plane and the velocity in z- direction is very small it could be considered as a quasi-2D motion in a short period. Therefore, we keep using the kinetic energy format (with a coefficient $\frac{1}{2}$) to facilitate comparison between the translational and rotational modes.

The director $\mathbf{m}(t)$ can be used to obtain the angular velocity. Specifically, the i -th component of $d\mathbf{m}/dt$ is $\dot{\mathbf{m}}_i = (\boldsymbol{\omega} \times \mathbf{m})_i = \epsilon_{ijk}\omega_j m_k$. Here ϵ_{ijk} is the Levi-Civita symbol and Einstein summation is used. Hence, $|\dot{\mathbf{m}}|^2 = \dot{m}_i \dot{m}_i = \epsilon_{ijk}\epsilon_{ipq}\omega_j \omega_p m_k m_q = (\delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp})\omega_j \omega_p m_k m_q = \omega_j \omega_j m_k m_k - \omega_j m_j \omega_p m_p$. Since $m_k m_k = 1$ and $\omega_j m_j = \omega_{\parallel}$ (the angular velocity component in the \mathbf{m} direction), $|\dot{\mathbf{m}}|^2 = \omega^2 - \omega_{\parallel}^2 = \omega_{\perp}^2$. This result means that the director \mathbf{m} is related to the angular component transverse to the particle axis. Intuitively, it can be understood as the rotation along the cylinder axis \mathbf{m} does not change the director \mathbf{m} . Subsequently, a partial rotational energy (transpose to the cylinder axis) $E_{R\perp} = \langle \frac{1}{2} J \omega_{\perp}^2 \rangle$ can be obtained (FIG 36 c), where $J = \frac{1}{4}\mu r^2 + \frac{1}{12}\mu l^2 = \frac{7}{48}\mu l^2$ is the moment of inertial for the sample cylinder rotating along a transverse axis, $l = 12.5$ mm is the cylinder height and $r = \frac{1}{2}l$.





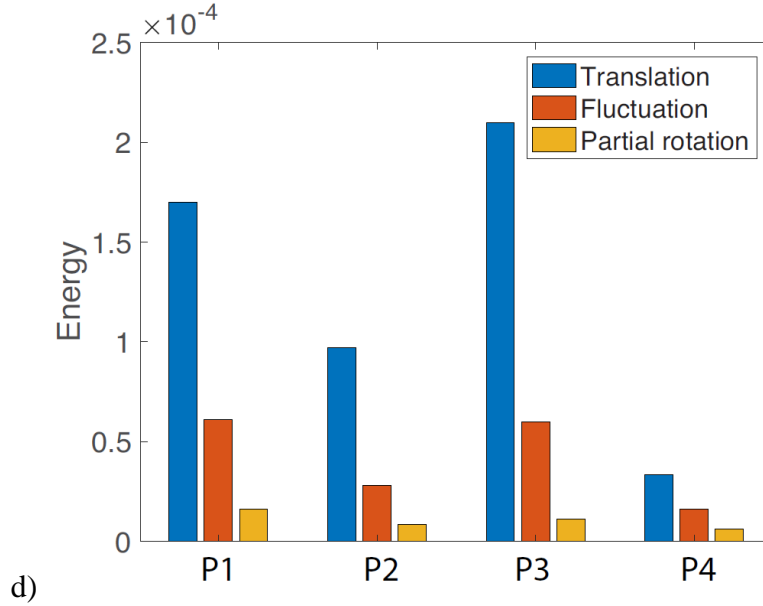


Figure 36. a) The distribution of kinetic energy per unit mass E_T/μ , b) E'_T/μ , c) $E_{R\perp}/\mu$

The distributions of the average translational energy per unit mass (E_T/μ) and partial rotational energy ($E_{R\perp}/\mu$) are shown in FIG 36 a and b. Both of them increase significantly along the radial direction. The rotational energy also has strong gradient near the surface, where the sample particle has less constraint. In the bulk flow, the magnitudes of E_T and $E_{R\perp}$ are comparable at a given spatial point P4. This suggests that particle rotation in a Couette Cell can be critical to the dynamics of a granular flow.

We also note that the large gradient of translational energy occurs at a large radius and lower height, while the large gradient of rotational energy occurs at the highest height across the entire radius. This indicates that the neighboring grains have more constraint on tracer particle for rotational movement than the translational one. Because at the top it is the transition zone from free surface to immerse in granular material, the large gradient means the granular material around it absorbs its energy rapidly, so the rotational energy of the tracer particle drops significantly. The

first thing we notice about FIG. 36 d) is that the magnitude of E_T, E'_T and $E_{R\perp}$ are significantly smaller at inner radius (P4) indicating the radial direction plays a key role in the energy distribution in the Couette cell system; while the impact of height is not that strong as the energy at P1 and P3 does not vary too much. FIG. 36 d) shows the translational energy is the most dominant one when the tracer particle is not too close to the central axis. It is evident that the translational energy varies greatly within different radius at the same height, and it follows the same trend at each height that the further the tracer particle moves away from the center of Couette cell, the greater the translational energy. However, the rotational energy is more stable across the different radius.

We also analyzed the independency of flipping and rotating motions. The probability of flipping $P(\text{flipping})$ and jumping $P(\text{jumping})$ from the symbolic data are 0.198 and 0.0242 respectively which indicate the percentage of time for flipping and jumping during the whole tracking process. The flipping process covers nearly 20% of the whole process while the jumping process rarely occurs indicates that the tracer particle is restricted in the radial direction for most of the time and within the restricted band it can rotate about its axis. Using Bayes' theorem, $P(\text{flipping}|\text{jumping}) = 0.3036$ which is approximately 50% higher than the $P(\text{flipping})$, and $P(\text{flipping}|\text{No jumping}) = 0.1975$ is almost identical as $P(\text{flipping})$. We also calculated the conditional probability $P(\text{jumping}|\text{flipping}) = 0.0372$ and $P(\text{jumping}|\text{No flipping}) = 0.0213$. These evident that the flipping process occurs regardless the existence of the jumping but the jumping status is definitely a stimulant for flipping to occur. And vice versa the flipping process increased the chance of jumping by 53% while if there is no flipping happening there is a slightly depression for jumping status. Overall, the flipping and jumping status are incentive for each other but the absence of either one has little effect on the other.

6 Conclusion

A novel experimental technology, magnetic particle tracking (MPT), is tested and compared with the optical method regarding the accuracy. It is proved that MPT is a solid method to reconstruct the trajectory and orientation of a magnetic particle. The accuracy for translational error is less than 0.5% and the rotation error is less than 1.5° . Because of the simplicity of the experiment device and free from radioactive material, it is a powerful tool for study flows in an opaque environment. By comparing the speed and accuracy of the optimization method, EKF method, and Particle Filter method, we find that the EKF algorithm is an appropriate algorithm for this problem. It provides both satisfactory accuracy and processing speed.

This technique is used to examine the flow in a split bottom Couette cell for different purposes. The MPT technology can provide the Lagrangian trajectory and orientation history of one sample particle over the entire measurement time. The trajectory of the sample particle shows that, in our setup, the bulk flow possesses a layered structure. The layers can be observed in some shear zone of monodispersed particulate flow or in the motion of large grains in a confined space. In layered flows, the anisotropic particle with a low aspect ratio has preferred alignment, which is largely due to the geometry shape and layer confinement.

The cylinder particle flips among different orientations and jumps between the layers. The duration of being in a preferred layer is longer than being in a preferred orientation, and both are longer than the time of transition. The transition is a highly random and intermittent process, but the order of director flipping is usually determined by the shear direction. Usually, jumping between layers is accompanied by director flipping, but flipping may occur in the same layer without jumping.

The rotational energy of the individual particle is approximately the same level as the translational energy near the center of the Couette cell, and the confinement of neighbor grains play a dominant role in restricting this rotational movement. The translational velocity energy varies significantly regarding the distance to the center of the Couette Cell while the rotational velocity energy is more stable inner the Couette Cell except for the upper free surface. The flipping and jumping process are stimulus to each other however the absence of either one has little effect on the other.

7 Reference

- [1] V. Pizzella *et al.*, "A novel method for real-time magnetic marker monitoring in the gastrointestinal tract," *Phys. Med. Biol.* 45, 2000.
- [2] S. Schaffelhofer, H. Scherberger, N. C. Atuegwu, and R. L. Galloway, "Accuracy assessment protocols for electromagnetic tracking systems," *Phys. Med. Biol.*, no. 48 (2003) 2241–2251, 2003.
- [3] Chao Hu, "Localization and Orientation System for Robotic Wireless Capsule Endoscope," *Nature*, vol. 405, no. 6785, 2006.
- [4] W. Andrä *et al.*, "A novel method for real-time magnetic marker monitoring in the gastrointestinal tract.," *Phys. Med. Biol.*, vol. 45, no. 10, pp. 3081–3093, 2000.
- [5] C. Hu, M. Li, S. Song, W. Yang, R. Zhang, and M. Q. Meng, "A Cubic 3-Axis Magnetic Sensor Array for Wirelessly Tracking Magnet Position and Orientation," vol. 10, no. 5, pp. 903–913, 2010.
- [6] V. Schlageter, P. Besse, R. S. Popovic, and P. Kucera, "Tracking system with ⁶ve degrees of freedom using a 2D-array of Hall sensors and a permanent magnet," vol. 92, pp. 37–42, 2001.
- [7] S. F. E. GOLDEN ROBERT N, "APPARATUS AND METHOD FOR LOCATING A MEDICAL TUBE IN THE BODY OF A PATIENT," CA19942171717 19940914.
- [8] H. Wensheng, Z. Xiaolin, and P. Chenglin, "Experimental study of magnetic-based localization model for miniature medical device placed indwelling human body," *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 7 VOLS, no. 2, pp. 1309–1312, 2005.
- [9] M. Q.-. M. and Y. C. Xiaona Wang, "A low-cost tracking method based on magnetic marker for capsule endoscope," in *International Conference on Information Acquisition, 2004. Proceedings, 2004*, pp. 524–526.
- [10] S. Yamada, C. P. Gooneratne, M. Iwahara, and M. Kakikawa, "Detection and estimation of low-concentration magnetic fluid inside body by a needle-type GMR sensor," *IEEE Trans. Magn.*, vol. 44, no. 11 PART 2, pp. 4541–4544, 2008.
- [11] C. Carr, A. N. Matlachov, H. Sandin, M. A. Espy, and R. H. Kraus, "Magnetic sensors for bioassay: HTS SQUIDS or GMRs?," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 808–811, 2007.
- [12] J. Daughton, "GMR and SDT sensors," *Dig. Intermag Conf.*, vol. 36, no. 5, pp. 2773–2778, 2000.
- [13] E. E. Patterson, J. Halow, and S. Daw, "Innovative Method Using Magnetic Particle Tracking to Measure Solids Circulation in a Spouted Fluidized Bed," pp. 5037–5043, 2010.
- [14] K. A. Buist, T. W. Van Erdewijk, N. G. Deen, and J. A. M. Kuipers, "Determination and Comparison of Rotational Velocity in a Pseudo 2-D Fluidized Bed Using Magnetic Particle Tracking and Discrete Particle Modeling," vol. 61, no. 10, 2015.
- [15] and J. A. M. K. Kay A. Buist, Alex C. van der Gaag, Niels G. Deen, "Improved Magnetic Particle Tracking Technique in Dense Gas Fluidized Beds Kay," *AIChE J.*, vol. 56, no. 4, pp. 1495–1502, 2014.
- [16] L. Yang, J. T. Padding, K. A. Buist, and J. A. M. Kuipers, "Three-dimensional fluidized beds with rough spheres: Validation of a Two Fluid Model by Magnetic Particle Tracking and discrete

- particle simulations,” *Chem. Eng. Sci.*, vol. 174, no. September, pp. 238–258, 2017.
- [17] I. Mema, K. A. Buist, J. A. M. Kuipers, and J. T. Padding, “Fluidization of spherical versus elongated particles: Experimental investigation using magnetic particle tracking,” *AIChE J.*, no. December 2019, pp. 1–13, 2019.
- [18] E. Sette, D. Pallarès, F. Johnsson, F. Ahrentorp, A. Ericsson, and C. Johansson, “Magnetic tracer-particle tracking in a fluid dynamically down-scaled bubbling fluidized bed,” *Fuel Process. Technol.*, vol. 138, pp. 368–377, 2015.
- [19] F. Charru, B. Andreotti, and P. Claudin, “Sand ripples and dunes,” *Annu. Rev. Fluid Mech.*, vol. 45, pp. 469–493, 2013.
- [20] F. Toschi and E. Bodenschatz, “Lagrangian properties of particles in turbulence,” *Annu. Rev. Fluid Mech.*, vol. 41, pp. 375–404, 2009.
- [21] N. Chenouard *et al.*, “Objective comparison of particle tracking methods,” *Nat. Methods*, vol. 11, no. 3, pp. 281–289, 2014.
- [22] A. Amon *et al.*, “Preface: Focus on imaging methods in granular physics,” *Rev. Sci. Instrum.*, vol. 88, p. 051701, 2017.
- [23] N. T. Ouellette, H. Xu, and E. Bodenschatz, “A quantitative study of three-dimensional Lagrangian particle tracking algorithms,” *Exp. Fluids*, vol. 40, pp. 301–313, 2006.
- [24] C. Shen *et al.*, “Brain-Like Navigation Scheme based on MEMS-INS and Place Recognition,” *Appl. Sci.*, vol. 9, no. 8, p. 1708, 2019.
- [25] C. Shen, J. Yang, J. Tang, J. Liu, and H. Cao, “Note: Parallel processing algorithm of temperature and noise error for micro-electro-mechanical system gyroscope based on variational mode decomposition and augmented nonlinear differentiator,” *Rev. Sci. Instrum.*, vol. 89, no. 7, 2018.
- [26] X. Guo, J. Tang, J. Li, C. Shen, and J. Liu, “Attitude measurement based on imaging ray tracking model and orthographic projection with iteration algorithm,” *ISA Trans.*, no. xxx, 2019.
- [27] X. Guo, J. Tang, J. Li, C. Wang, C. Shen, and J. Liu, “Determine turntable coordinate system considering its non-orthogonality,” *Rev. Sci. Instrum.*, vol. 90, no. 3, p. 033704, Mar. 2019.
- [28] J. S. Lin, M. M. Chen, and B. T. Chao, “A novel radioactive particle tracking facility for measurement of solids motion in gas fluidized beds,” *AIChE J.*, vol. 31, no. 3, pp. 465–473, 1985.
- [29] D. . Parker *et al.*, “Developments in particle tracking using the Birmingham Positron Camera,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 392, no. 1–3, pp. 421–426, Jun. 1997.
- [30] D. . Parker, R. . Forster, P. Fowles, and P. . Takhar, “Positron emission particle tracking using the new Birmingham positron camera,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 477, no. 1–3, pp. 540–545, Jan. 2002.
- [31] D. J. Parker, M. R. Hawkesworth, C. J. Broadbent, P. Fowles, T. D. Fryer, and P. A. McNeil, “Industrial positron-based imaging: Principles and applications,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 348, no. 2–3, pp. 583–592, Sep. 1994.

- [32] D. J. Parker, C. J. Broadbent, P. Fowles, M. R. Hawkesworth, and P. McNeil, "Positron emission particle tracking - a technique for studying flow within engineering equipment," *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 326, no. 3, pp. 592–607, Mar. 1993.
- [33] R. D. Wildman and D. J. Parker, "Coexistence of Two Granular Temperatures in Binary Vibrofluidized Beds," *Phys. Rev. Lett.*, vol. 88, no. 6, p. 064301, 2002.
- [34] J. B. Romero and D. W. Smith, "Flash x-ray analysis of fluidized beds," *AIChE J.*, vol. 11, no. 4, pp. 595–600, 1965.
- [35] E. W. Grohse, "Analysis of gas-fluidized solid systems by x-ray absorption," *AIChE J.*, vol. 1, no. 3, pp. 358–365, 1955.
- [36] X. Chen, W. Zhong, and T. J. Heindel, "Using stereo XPTV to determine cylindrical particle distribution and velocity in a binary fluidized bed," *2AIChE*, vol. 65, no. 2, pp. 520–535, 2019.
- [37] T. J. Heindel, "A Review of X-Ray Flow Visualization With Applications to Multiphase Flows," *J. Fluids Eng.*, vol. 133, no. 7, p. 074001, 2011.
- [38] H. Nadeem and T. J. Heindel, "Review of noninvasive methods to characterize granular mixing," *Powder Technol.*, vol. 332, pp. 331–350, 2018.
- [39] C. Scholz and T. Pöschel, "Velocity distribution of a homogeneously driven two-dimensional granular gas," *Phys. Rev. Lett.*, vol. 118, p. 198003, 2017.
- [40] X. Tao and H. Wu, "A Comparison of the Sequential Quadratic Programming Algorithm and Extended Kalman Filter Method in the Magnetic Particle Tracking Reconstruction," no. January, pp. 1–10, 2019.
- [41] G. Mohs, O. Gryczka, and S. Heinrich, "Chemical Engineering Science," vol. 64, 2009.
- [42] V. Idakiev and L. Mörl, "How to measure the particle translation and rotation in a spouted and fluidized bed?," *J. Chem. Technol. Metall.*, vol. 48, no. 5, pp. 445–450, 2013.
- [43] A. Köhler, D. Pallarès, and F. Johnsson, "Magnetic tracking of a fuel particle in a fluid-dynamically down-scaled fluidised bed," *Fuel Process. Technol.*, vol. 162, pp. 147–156, 2017.
- [44] A. Köhler, A. Rasch, D. Pallarès, and F. Johnsson, "Experimental characterization of axial fuel mixing in fluidized beds by magnetic particle tracking," *Powder Technol.*, vol. 316, pp. 492–499, 2017.
- [45] L. Zhang *et al.*, "Experimental study of the particle motion in flighted rotating drums by means of Magnetic Particle Tracking," *Powder Technol.*, vol. 339, pp. 817–826, 2018.
- [46] P. Schall and M. van Hecke, "Shear Bands in Matter with Granularity," *Annu. Rev. Fluid Mech.*, vol. 42, pp. 67–88, 2010.
- [47] T. Unger, "Refraction of shear zones in granular materials," *Phys. Rev. Lett.*, vol. 98, p. 018301, 2007.
- [48] K. Nichol, A. Zanin, R. Bastien, E. Wandersman, and M. Van Hecke, "Flow-induced agitations create a granular fluid," *Phys. Rev. Lett.*, vol. 104, p. 078302, 2010.
- [49] D. L. Henann and K. Kamrin, "Continuum modeling of secondary rheology in dense granular

- materials," *Phys. Rev. Lett.*, vol. 113, p. 178001, 2014.
- [50] G. Wortel, O. Dauchot, and M. Van Hecke, "Criticality in Vibrated Frictional Flows at a Finite Strain Rate," *Phys. Rev. Lett.*, vol. 117, p. 198002, 2016.
- [51] B. RA, "Experiments on a gravity-free dispersion of large solid spheres in a Newtonian fluid under shear," *Proc. R. Soc. London. Ser. A. Math. Phys. Sci.*, vol. 225, no. 1160, pp. 49–63, 1954.
- [52] M. L. Hunt, R. Zenit, C. S. Campbell, and C. E. Brennen, "Revisiting the 1954 suspension experiments of R. A. Bagnold," *J. Fluid Mech.*, vol. 452, pp. 1–24, 2002.
- [53] F. Da Cruz, S. Emam, M. Prochnow, J. N. Roux, and F. Chevoir, "Rheophysics of dense granular materials: Discrete simulation of plane shear flows," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 72, p. 021309, 2005.
- [54] F. Tapia, S. Shaikh, J. E. Butler, O. Pouliquen, and É. Guazzelli, "Rheology of concentrated suspensions of non-colloidal rigid fibres," *J. Fluid Mech.*, vol. 827, pp. 1–11, 2017.
- [55] F. Boyer, É. Guazzelli, and O. Pouliquen, "Unifying suspension and granular rheology," *Phys. Rev. Lett.*, vol. 107, p. 188301, 2011.
- [56] R. Maurin, J. Chauchat, and P. Frey, "Dense granular flow rheology in turbulent bedload transport," *J. Fluid Mech.*, vol. 804, pp. 490–512, 2016.
- [57] D. M. Mueth, G. F. Debregeas, P. J. Eng, S. R. Nagel, and H. M. Jaeger, "Signatures of granular microstructure in dense shear flows," *Nature*, vol. 406, pp. 385–389, 2000.
- [58] J. A. Dijksman and M. Van Hecke, "Granular flows in split-bottom geometries," *Soft Matter*, vol. 6, no. 13, pp. 2901–2907, 2010.
- [59] K. Sakaie, D. Fenistein, T. J. Carroll, M. Van Hecke, and P. Umbanhowar, "MR imaging of Reynolds dilatancy in the bulk of smooth granular flows," *Eur. Lett.*, vol. 84, no. 3, p. 38001, 2008.
- [60] D. Fenistein, J. W. Van De Meent, and M. Van Hecke, "Core precession and global modes in granular bulk flow," *Phys. Rev. Lett.*, vol. 96, p. 118001, 2006.
- [61] D. M. Mueth, "Measurements of particle dynamics in slow , dense granular Couette flow," no. March 2001, pp. 1–13, 2003.
- [62] R. Stannarius, "Magnetic Resonance Imaging of Granular Materials," vol. 88, p. 051806, 2017.
- [63] S. Wegner *et al.*, "Effects of grain shape on packing and dilatancy of sheared granular materials," *Soft Matter*, vol. 10, no. 28, pp. 5157–5167, 2014.
- [64] D. Fenistein and M. van Hecke, "Kinematics: Wide shear zones in granular bulk flow," *Nature*, vol. 425, no. 6955, p. 256, 2003.
- [65] S. Wegner, T. Börzsönyi, T. Bien, G. Rose, and R. Stannarius, "Alignment and dynamics of elongated cylinders under shear," *Soft Matter*, vol. 8, no. 42, pp. 10950–10958, 2012.
- [66] Y. Guo, C. Wassgren, W. Ketterhagen, B. Hancock, B. James, and J. Curtis, "A numerical study of granular shear flows of rod-like particles using the discrete element method," *J. Fluid Mech.*, vol. 713, pp. 1–26, 2012.
- [67] T. Börzsönyi *et al.*, "Shear-induced alignment and dynamics of elongated granular particles,"

- Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 86, p. 051304, 2012.
- [68] X. Tao, X. Tu, and H. Wu, "A new development in magnetic particle tracking technology and its application in a sheared dense granular flow," *Rev. Sci. Instrum.*, vol. 90, no. 6, p. 065116, 2019.
- [69] J. D. Jackson, *Classical Electrodynamics*, 3rd ed. Wiley, 1998.
- [70] K. A. Buist, A. C. van der Gaag, N. G. Deen, and J. A. M. Kuipers, "Improved magnetic particle tracking technique in dense gas fluidized beds," *AIChE*, vol. 60, no. 9, pp. 3133–3142, 2014.
- [71] J. M. Taylor *et al.*, "High-sensitivity diamond magnetometer with nanoscale resolution," *Nat. Phys.*, vol. 4, no. 10, pp. 810–816, 2008.
- [72] P. Del Moral, "MEASURE-VALUED PROCESSES AND INTERACTING PARTICLE SYSTEMS. APPLICATION TO NONLINEAR FILTERING PROBLEMS," *Ann. Appl. Probab.*, vol. 8, no. 2, pp. 438–495, 1998.
- [73] P. D. MORAL, "Nonlinear Filtering: Interacting Particle Resolution," *Markov Process. Relat. Fields*, vol. 2 (4): 555.
- [74] *Sequential Monte Carlo Methods in Practice*. 2001.
- [75] A. Doucet, N. Freitas, and N. Gordon, "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice*, 2001.
- [76] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking," vol. 50, no. 2, pp. 174–188, 2002.
- [77] Y. Shi and H. Fang, "Kalman filter-based identification for systems with randomly missing measurements in a network environment," *Int. J. Control*, vol. 83, no. 3, pp. 538–551, 2010.
- [78] F. Gustafsson, *Adaptive filtering and change detection*. New York: Wiley, 2000.
- [79] A. Smith, *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.
- [80] A. Gelb, *Applied optimal estimation*. Cambridge: MIT Press, 1974.
- [81] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. basic Eng.*, vol. 83, no. 1, pp. 95–108, 1961.
- [82] A. J. Chorin and X. Tu, "Implicit sampling for particle filters," *Proc. Natl. Acad. Sci.*, vol. 106, no. 41, pp. 17249–17254, 2009.
- [83] R. G. et Al., *Measurements of Turbulent Flows. In: Tropea C., Yarin A.L., Foss J.F. (eds) Springer Handbook of Experimental Fluid Mechanics*. 2007.
- [84] D. M. Mueth, G. F. Debregeas, and G. S. Karczmar, "Signatures of granular microstructure in dense shear flows," vol. 3, pp. 385–389, 2000.
- [85] X. Cheng *et al.*, "Three-Dimensional Shear in Granular Flow," vol. 038001, no. January, pp. 1–4, 2006.
- [86] D. Fenistein and M. Van Hecke, "Wide shear zones in granular bulk flow," vol. 425, no. SEPTEMBER, p. 99775, 2003.
- [87] K. Sakaie, D. Fenistein, T. J. Carroll, M. van Hecke, and P. Umbanhowar, "MR imaging of Reynolds

- dilatancy in the bulk of smooth granular flows," *EPL (Europhysics Lett.)*, vol. 84, no. 3, p. 38001, Nov. 2008.
- [88] G. P. Romano *et al.*, "Chapter 10 Measurements of turbulent flows," in *Handbook of Experimental Fluid Mechanics*, C. Tropea, A. L. Yarin, and J. F. Foss, Eds. Springer, 2007, pp. 754–856.
- [89] H. Wu, H. Xu, and E. Bodenschatz, "Measuring vorticity vector from the spinning of micro-sized mirror-encapsulated spherical particles in the flow," *Rev. Sci. Instrum.*, vol. 90, no. 11, p. 115111, 2019.
- [90] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Ann. Stat.*, vol. 38, no. 5, pp. 2916–2957, 2010.
- [91] H. Wu and Y. Li, "Mobility and volatility : What is behind the rising income inequality in the United States," *Physica A*, vol. 492, pp. 2345–2352, 2018.

8 Appendix

8.1 Magnetic Particle Tracking Code with Extended Kalman Filter Algorithm (MATLAB)

8.1.1 Read Me

This is a Magnetic Particle Tracking Package which trying to reconstruct trajectory and calculating the velocity of a magnetic tracer particle. This technique is created by Huixuan Wu (Advisor) and his first PhD student Xingtian Tao

1. Run the “[magnetmoment](#)” when you are using a new tracer particle to determine what is the Magnet moment and once you get the value you have to manually set it in the “[Kalman_MPT_main](#)” code and “[my_lincon](#)”.
2. Once you know the magnet moment and start doing the experiment collect the data with either time series recorded or not (In the “[pre_process](#)” function able and disable the loading text part)
3. Run the “[pre_process](#)” code to subtract the offset value and mark the index of out of range value
4. Run the “[Kalman_MPT_main](#)” code (The “[SQP_reconstruction](#)” is for initialize the position and “[my_nlincon](#)” is the constraint of the magnet moment)
5. Once you got the trajectory, run the “[Differentialkernel](#)” and “[Angular](#)” for velocity and angular velocity respectively. Note that: the “[Differentialkernel](#)” is timeless unit.
6. the “[Rotationbowl](#)” and “[Couette_cell_Analysis](#)” is for Couette Cell distribution analysis may or may not useful for your project purpose.

8.1.2 Find the Magnet Moment

```
clear
u0=4*pi*1e-7;

Path='D:\KU\Research\magnet\newDAQ\New Sensor\cylindercali\';
Files1=dir([Path,'test2.lvm']);
Files2=dir([Path,'nomag.lvm']);
%%% sensortable is the position and orientation of the magnetometer
%%% it is obtained from calibration
sensortable=[0.0674 0 0 0 0 1
             0.0674 0 0 1 0 0
             0.0674 0 0 0 1 0
             -0.0674 0 0.0375 0 0 1
             -0.0674 0 0.0375 -1 0 0
             -0.0674 0 0.0375 0 -1 0
```

```

-0.0337 -0.05837 0 0 0 1
-0.0337 -0.05837 0 -0.5 -0.866 0
-0.0337 -0.05837 0 0.866 -0.5 0
0.0337 0.05837 0.0375 0 0 1
0.0337 0.05837 0.0375 0.5 0.866 0
0.0337 0.05837 0.0375 -0.866 0.5 0];
chip=sensortable(:,1:3);
Mag=sensortable(:,4:6);

fileID2=fopen([Path, '/', Files2.name]);

B=textscan(fileID2, '%s %f %f %f %f %f %f %f %f %f %f %f %f %f', 'HeaderLines', 50)';%
L=length(B{1});

BB=[B{2}(1:L,1), B{3}(1:L,1), B{4}(1:L,1), B{5}(1:L,1), B{6}(1:L,1), B{7}(1:L,1), B{8}(1:L,1) ...
B{9}(1:L,1), B{10}(1:L,1), B{11}(1:L,1), B{12}(1:L,1), B{13}(1:L,1), B{14}(1:L,1)]
;
fileID1=fopen([Path, '/', Files1.name]);

A=textscan(fileID1, '%s %f %f %f %f %f %f %f %f %f %f %f %f %f', 'HeaderLines', 50)';%
L=length(A{1});

AA=[A{2}(1:L,1), A{3}(1:L,1), A{4}(1:L,1), A{5}(1:L,1), A{6}(1:L,1), A{7}(1:L,1), A{8}(1:L,1) ...
A{9}(1:L,1), A{10}(1:L,1), A{11}(1:L,1), A{12}(1:L,1), A{13}(1:L,1), A{14}(1:L,1)]
;
nomag=mean(BB); % + mean(CC) / 2;

for i=1:length(AA)
datawithnoise(i, :)=(AA(i, :)-nomag)*1000;
end

%%% this is the initial guess %%%
X(1, :)= [0, 0, 0.02, 0.010, 0.01, 0];

%%% lower bound and upper bound of test domain and the strength of magnet
%%% moment
LB=[-0.05, -0.05, 0, -0.015, -0.015, -0.015];
UB=[0.05, 0.05, 0.1, 0.015, 0.015, 0.015];
% tic
options = optimoptions('fmincon', 'Algorithm', 'sqp');
f=@(x) Qquality(x, datawithnoise(1, :), sensortable);
problem1 =
createOptimProblem('fmincon', 'objective', f, 'x0', X(1, :), 'lb', LB, 'ub', UB, 'options', options);

gs=GlobalSearch('Display', 'off', 'NumStageOnePoints', 500, 'NumTrialPoints', 3e3, 'StartPointsToRun', 'bounds-ineqs');

gs2=GlobalSearch('Display', 'off', 'NumStageOnePoints', 500, 'NumTrialPoints', 3e3, 'StartPointsToRun', 'bounds-ineqs')
[position1(1, :), fvalinit, exitflag(1), output]=run(gs, problem1);

```



```

% Process noise covariance and initial guess
S_aposteriori = [mean(Initial_Pos(3:6,:))]; % initial position
estimation ,0,0,0,0,0,0 position(1,:);%
P_aposteriori = 2e-8*eye(length(S_aposteriori)); %initial variation
estimation
Q=3e-9*eye(length(S_aposteriori)); % motion prediction error
% Q(4:6,4:6)=3*Q(4:6,4:6);
R=[5e-3 1e-5]; % sensor noise, first is the percentage, second is the offset
% R(1)=1
%% ===== reconstruction
Recons_position=EKF_MPT(data_noise,sensortable, dt,Moment,Q, R,
P_aposteriori,S_aposteriori,out_rang_index);
% Recons_position=UKF_MPT(data_noise,sensortable, dt,Moment,Q, R,
P_aposteriori,S_aposteriori);
Recons_position1=smoothdata(Recons_position,'gaussian',25);

% ----- end of main program

```

8.1.5 SQP Algorithm for Initial Guess

```

function [position1, fval, exitflag]=SQP_reconstruction(data,sensor, LB, UB)

u0=4*pi*1e-1;
[M,N]=size(data);
%% initial arbitrary guess
X=[-0.03,0,0.05,0.0010,0.0003,0];
options = optimoptions('fmincon','Algorithm','sqp');
f=@(x)Qquality(x,data(1,:),sensor);
problem1 =
createOptimProblem('fmincon','objective',f,'x0',X,'lb',LB,'ub',UB,'nonlcon',@
my_nlincon,'options',options);

gs=GlobalSearch('Display','off','NumStageOnePoints',1000,'NumTrialPoints',1e3
,'StartPointsToRun','bounds-ineqs');

gs2=GlobalSearch('Display','off','NumStageOnePoints',500,'NumTrialPoints',1e3
,'StartPointsToRun','bounds-ineqs');
[position1(1,:),fvalinit,exitflag(1),output]=run(gs,problem1);

for j=2:M
f=@(x)Qquality(x,data(j,:),sensor);
problem2 = createOptimProblem('fmincon','objective',f,'x0',position1(j-
1,:), 'lb',LB,'ub',UB,'nonlcon',@my_nlincon,'options',options);
[position1(j,:),fval(j),exitflag(j),output]=run(gs2,problem2);

End

```

8.1.6 Constrains of SQP algorithm

```

function [c,ceq] = my_nlincon(x)
Moment=0.0120;
c=[];

```

```
ceq=sqrt(x(4)^2+x(5)^2+x(6)^2)-Moment;
end
```

8.1.7 Extended Kalman Filter Algorithm

```
function Recons_position=EKF_MPT(data,sensortable, dt,Moment,Q, R,
P_aposteriori,S_aposteriori,out_rangeindex)
N=length(data);

Recons_position=zeros(N,length(S_aposteriori));
Recons_position(1,:)=S_aposteriori;

for k=2:N
    H=zeros;
    data2=data(k,:);
    index=find(~out_rangeindex(k,:));
    %   if isempty(index)==1
    %       continue
    %   else
    data2(out_rangeindex(k,:)==0)=[];
    %   end

    if length(S_aposteriori)==12
        % update velocity
        Ux=S_aposteriori(7);
        Uy=S_aposteriori(8);
        Uz=S_aposteriori(9);
        Omega=S_aposteriori(10);
        theta=S_aposteriori(11);
        phi=S_aposteriori(12);

        %Rotation matrix
        W=[0 -cos(theta) sin(theta)*sin(phi);cos(theta) 0 -sin(theta)*cos(phi);-
sin(theta)*sin(phi) sin(theta)*cos(phi) 0];
        Rot=eye(3)+sin(Omega*dt)*W+(1-cos(Omega*dt))*W*W;

        % Update apriori estimate
        DX1=[Ux,Uy,Uz]*dt;
        S_apriori(1:3) = S_aposteriori(1:3) + DX1;% need to restrict the
momentposition(k-1,1:6)
        S_apriori(4:6) = Rot*S_aposteriori(4:6)';

        S_apriori(7:12) = S_aposteriori(7:12);

        % Jacobians
        A = [eye(3),zeros(3), dt*eye(3),zeros(3);...
zeros(3), Rot , zeros(3), zeros(3);...dRot_dW(Rot,W, Omega, theta,
phi, S_aposteriori(4:6)', dt)
zeros(3),zeros(3),eye(3),zeros(3);...
zeros(3),zeros(3),zeros(3),eye(3)];
    elseif length(S_aposteriori)==6
        S_apriori = S_aposteriori;
        A=eye(6);
    end
end
```

```

P_apriori = A*P_aposteriori*A' + Q;%

Z_apriori=Mag_dipole(S_apriori, sensortable);
Z_apriori(out_rangeindex(k,')==0)=[];

Measure_R = (R(1)*data2(:)).^2+R(2);
[H, HPHR] = Kalman_H_generator(S_apriori, sensortable, P_apriori,
Measure_R, length(S_aposteriori),index);
% K = P_apriori*H'/(H*P_apriori*H'+ R);
K = P_apriori*H'*inv(HPHR);

correction = K*(data2'-Z_apriori);
% correction = K*(data(k,:)'-Z_apriori);
P_aposteriori = (eye(length(S_aposteriori))-K*H)*P_apriori;
S_aposteriori = S_apriori+correction';
S_aposteriori(4:6) = S_aposteriori(4:6)/norm(S_aposteriori(4:6))*Moment;

Recons_position(k,:)=S_aposteriori;

% if mod(k,20)==0 %& k >100
% correction
% end

end

```

8.1.8 Kalman Gain Generator

```

function [H,HPHR,flag] = Kalman_H_generator(S,
sensortable,P,Measure_R,ss,index)
[m,n]=size(sensortable);
H=zeros(m-length(index), ss);
% flag=logical(ones(1,m));

X=S(1:3);
Moment=S(4:6);

for i=1:m

X0=sensortable(i,1:3);
R=(X-X0);
rr=norm(R);
e=sensortable(i,4:6);
A=Moment*R';
B=R*e';

for j=1:3
H(i,j)=3*((Moment(j)*B+A*e(j))/rr^5-
5*A*B*R(j)/rr^7)+3*Moment*e'*R(j)/rr^5;
end
for j=4:6
H(i,j)=3*R(j-3)*B/rr^5-e(j-3)/rr^3;
end

% if norm(H(i,:))>1e3
% flag(i)=0;

```

```

%     end

end
H(index,:)=[];

H=H*1E-1;
HPHR=H*P*H' + diag(Measure_R);

```

8.1.9 Velocity Kernel

```

function [velocity, accler]= Differentialkernel(x);

sum=0;
sum2=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is an arbitrary track using for testing the function
% for t=1:20;
% % a=7;
% % x(t)=5*t+0.5*a*t^2;
% fun=@(t) 2*t.^2-3*t+1;
% x(t)=integral(fun,0,t)
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x=[1 4 9 16];
for t=5:length(x)-5;
for tau=-4:4;
T=4;
W=T/3.3;

% (x(1)-tau).*Aa*exp(-T.^2/W^2)*(1-2*tau.^2/W.^2)+Ba;
Av=(0.5*W^2*(W*sqrt(pi)*erf(T/W)-2*T*exp(-T^2/W^2)))^(-1);
Kv=Av*tau*exp(-tau^2/W^2);
sum=sum+x(t+tau)*Kv;
Aa=(0.5*sqrt(pi)*erf(T/W).*W^3+exp(-T^2/W^2)*(1/3-T*W^2-T^3))^(-1);
Ba=-Aa.*exp(-T.^2/W.^2);
% sum=sum+(x(T)-tau).*Aa*exp(-T.^2/W^2)*(1-2*T.^2/W.^2)+Ba;
Ka=Aa*exp(-tau^2/W^2)*(1-2*tau^2/W^2)+Ba;
sum2=sum2+x(t+tau)*Ka;
end
velocity(t)=sum;
sum=0;
accler(t)=-sum2;
sum2=0;
end

```

8.1.10 Angular Velocity

```

function [angular_v]=Angular(moment,dt,N)
x=moment;
% sum2=0;
ang_vel=zeros;
v=zeros(5,3);
axis=zeros(4,3);

```

```

for t=1:length(x)-N-1
    sum=0;
%% find the plane
% x=[1 0 0;0 1 0;2 1 0;1 2 0;1 1 0]
    p=mean(x(t:N+t,:),1);
    R=bsxfun(@minus,x(t:N+t,:),p);
    [V,D]=eig(R'*R);
    n=V(:,1);
    n=n/norm(n)*0.0105; %%%%%%%%% the normal vector to the plane
    V=V(:,2:end);
    hub(t,:)=n;
    rt(t,:)=x(t,:)-x(t,:)*n*n';
    phi(t)=acos(x(t,:)*hub(t,:)'/ (norm(x(t,:))*norm(hub(t,:))));
    r(t)=sin(phi(t))*0.0105;
    r3(t)=norm(x(t+1,:)-x(t,:));
end

for i=3:length(rt)-1
    if r3(i)>r(i)+r(i+1) || r(i+1)>r(i)+r3(i) || r(i)>r(i+1)+r3(i)
        continue
    end
    theta(i)=acos((r(i)^2+r(i+1)^2-r3(i)^2)/2/r(i)/r(i+1));
%     theta(i)=acos(rt(i,:)*rt(i+1,:)' /norm(rt(i,:))/norm(rt(i+1,:)));
end
theta=smooth(theta,N);
angular_v=theta/dt;
end

```

8.2 Magnetic Particle Tracking with Particle Filter Algorithm

8.2.1 Particle Filter Main Code

Note that: this code including 6 different noise level comparison and the virtual sample are generated separately

```

%% clear memory, screen, and close all figures
clear, clc, close all;
clear
rng(31415826);
gamma=0.1;
K=0.008;
dt=0.01;
v0=20;
w0=1;
T=300;

re_scale_f=7000;

Hx=v0*(rand(T,1)-0.5);
Hy=v0*(rand(T,1)-0.5);
Hz=v0*(rand(T,1)-0.5);
HRx=w0*(rand(T,1)-0.5);

```

```

HRy=w0*(rand(T,1)-0.5);
HRz=w0*(rand(T,1)-0.5);
Duration=10*round(30*rand(T,1));

Steps=sum(Duration)+1;

X=zeros(Steps, 1);
Ux=X;
Wx=X;
Y=zeros(Steps, 1);
Uy=X;
Wy=X;
Z=zeros(Steps, 1);
Uz=X;
Wz=X;
M=zeros(Steps, 3);
M(:,3)=1;
% P=zeros(Steps, 1);
% Q=zeros(Steps, 1);

s=1;

for j=1:T
    Ux(s)=Ux(s)+Hx(j);
    Uy(s)=Uy(s)+Hy(j);
    Uz(s)=Uz(s)+Hz(j);
    Wx(s)=Wx(s)+HRx(j);
    Wy(s)=Wy(s)+HRy(j);
    Wz(s)=Wz(s)+HRz(j);

    for k=1:Duration(j)
        s=s+1;
        % velocity and position updates, a gravity term is used to control
        % the particle range
        X(s)=X(s-1)+dt*Ux(s-1);
        Ux(s)=Ux(s-1)-dt*(gamma*Ux(s-1)+K*X(s-1));
        Y(s)=Y(s-1)+dt*Uy(s-1);
        Uy(s)=Uy(s-1)-dt*(gamma*Uy(s-1)+K*Y(s-1));
        Z(s)=Z(s-1)+dt*Uz(s-1);
        Uz(s)=Uz(s-1)-dt*(gamma*Uz(s-1)+K*Z(s-1));
        % angular velocity and rotation updates
        Wx(s)=Wx(s-1)-dt*gamma*Wx(s-1);
        Wy(s)=Wy(s-1)-dt*gamma*Wy(s-1);
        Wz(s)=Wz(s-1)-dt*gamma*Wz(s-1);

        tmp = M(s-1,:) - dt*cross(M(s-1,:), [Wx(s-1), Wy(s-1), Wz(s-1)]);

        M(s,:) = tmp/norm(tmp);
    end

end

% from mm to meter
X=X/re_scale_f;
Y=Y/re_scale_f;
Z=Z/re_scale_f+0.01;
Ux=Ux/re_scale_f;
Uy=Uy/re_scale_f;

```

```

Uz=Uz/re_scale_f;
M=M*0.0105;
X=X(1:10:end);
Y=Y(1:10:end);
Z=Z(1:10:end);
Ux=Ux(1:10:end);
Uy=Uy(1:10:end);
Uz=Uz(1:10:end);
M=M(1:10:end,:);
Wx=Wx(1:10:end);
Wy=Wy(1:10:end);
Wz=Wz(1:10:end);
sensortable=[0.0545  0  0.0109  0  0  1;      0.0545  0  0.0109  1  0  0;
              0.0545  0  0.0109  0  1  0;      -0.0546  0  0.0109  0  0  1;
              -0.0546  0  0.0109  -1  0  0;      -0.0546  0  0.0109  0  -1  0;
              0.05435  0.04883  0.0109  0  0  1;      0.05435  0.04883  0.0109  1
0  0;
              0.05435  0.04883  0.0109  0  1  0;      -0.04845  0.04772  0.0109
0  0  1;
              -0.04845  0.04772  0.0109  -1  0  0;      -0.04845  0.04772  0.0109  0
-1  0 ];
figure
quiver3(X,Y,Z,M(:,1),M(:,2),M(:,3))
hold on
plot3(sensortable(1,1),sensortable(1,2),sensortable(1,3),'ks')
plot3(sensortable(4,1),sensortable(4,2),sensortable(4,3),'ks')
plot3(sensortable(7,1),sensortable(7,2),sensortable(7,3),'ks')
plot3(sensortable(11,1),sensortable(11,2),sensortable(11,3),'ks')

Steps=length(X);
% signal based on the chip position
u0=4*pi*1e-1;

datareading=zeros(Steps,12);
for j=1:Steps
    for k=1:12
        r=[sensortable(k,1)-X(j), sensortable(k,2)-Y(j),sensortable(k,3)-
Z(j)];
        rr=norm(r);
        datareading(j,k)=u0/4/pi*...
            (3*r*(M(j,:)*r')/rr^5-
M(j,:)/rr^3)*sensortable(k,4:6) '*89;
    end
end

position=[X,Y,Z,M,Ux,Uy,Uz,Wx,Wy,Wz];
xx=X';
yy=Y';
zz=Z';
datawithnoise_one=datareading.*normrnd(1,0.01,[4693,12]);
datawithnoise_three=datareading.*normrnd(1,0.03,[4693,12]);
datawithnoise_six=datareading.*normrnd(1,0.06,[4693,12]);
datawithnoise_ten=datareading.*normrnd(1,0.1,[4693,12]);
datawithnoise_twenty=datareading.*normrnd(1,0.2,[4693,12]);

```

```

dt=0.005;
Moment=0.0105;
UB=[0.01,0.05,0.030,0.0105,0.0105,0.0105];
LB=[-0.04,-0.07,-0.01, -0.0105,-0.0105,-0.0105];
domain=UB-LB;
sensortable=[0.0545 0 0.0109 0 0 1;      0.0545 0 0.0109 1 0 0;
              0.0545 0 0.0109 0 1 0;      -0.0546 0 0.0109 0 0 1;
              -0.0546 0 0.0109 -1 0 0;      -0.0546 0 0.0109 0 -1 0;
              0.05435 0.04883 0.0109 0 0 1;      0.05435 0.04883 0.0109 1
0 0;
              0.05435 0.04883 0.0109 0 1 0;      -0.04845 0.04772 0.0109
0 0 1;
              -0.04845 0.04772 0.0109 -1 0 0;      -0.04845 0.04772 0.0109 0
-1 0 ];
%% Process equation x[k] = sys(k, x[k-1], u[k]);
nx = 6; % number of states
sigma_v=sqrt(1); sigma_w=sqrt(5);
% predictParticle = transitionstate(prevParticle);

p_obs_noise=@(v) normpdf(v,0,sigma_v);
gen_sys_noise = @(u) normrnd(0, sigma_w);
% obs=observation(Particlek);
p_yk_given_xk = @(yk,obs) p_obs_noise(yk - obs);
T=length(datareading);

gen_x0 = @(x) normrnd(0, sqrt(10));
p_sys_noise=@(w) normpdf(w,0,sigma_w);
Num_state=6;
% [Initial_Pos, fval, exitflag]=SQP_reconstruction(datareading(1:6,1:12),
sensortable, LB, UB);
% Initial_Pos=mean(Initial_Pos);
initialize=[-0.0,0.001,0.0100,0.00051,0.000244,0.00883];%[Initial_Pos(1:3),0
0,0,Initial_Pos(4:6),0,0,0];%[-0.0171,0.01,0.0151,0,0,0,0.0043,-0.0094,-
0.002,0,0,0];;

pf.k          = 2;          % initial iteration number
pf.Ns         = 5000;      % number of particles
pf.w          = zeros(pf.Ns, T); % weights
pf.particles  = zeros(Num_state, pf.Ns, T); % particles
pf.gen_x0     = gen_x0;    % function for sampling from
initial pdf p_x0
pf.p_yk_given_xk = p_yk_given_xk; % function of the observation
likelihood PDF p(y[k] | x[k])
pf.gen_sys_noise = gen_sys_noise; % function for generating system
noise
%pf.p_x0 = p_x0; % initial prior PDF p(x[0])
%pf.p_xk_given_xkm1 = p_xk_given_xkm1; % transition prior PDF p(x[k] |
x[k-1])
rng('default');
% robotPred = zeros(length(t),2);
% Recon_posi_pf = zeros(length(t),2);
% Num_particle=5000;
% initialPose =zeros;
prevPartilces(1,:) = initialize;
for k = 2:length(datareading)
    pf.k = k;

```

```

[xh_0(:,k), pf] = particle_filter(initialize,datareading(k,1:12), pf,
'systematic_resampling');

end
haha=1;
for k=2:length(datareading)
    pf.k = k;
    [xh_1(:,k), pf] = particle_filter(initialize,datawithnoise_one(k,1:12),
pf, 'systematic_resampling');
end

for k=2:length(datareading)
    pf.k = k;
    [xh_3(:,k), pf] =
particle_filter(initialize,datawithnoise_three(k,1:12), pf,
'systematic_resampling');
end

for k=2:length(datareading)
    pf.k = k;
    [xh_6(:,k), pf] = particle_filter(initialize,datawithnoise_six(k,1:12),
pf, 'systematic_resampling');
end

for k=2:length(datareading)
    pf.k = k;
    [xh_10(:,k), pf] = particle_filter(initialize,datawithnoise_ten(k,1:12),
pf, 'systematic_resampling');
end

for k=2:length(datareading)
    pf.k = k;
    [xh_20(:,k), pf] =
particle_filter(initialize,datawithnoise_twenty(k,1:12), pf,
'systematic_resampling');
end
%
save('PF_synthetic_positions.mat','xh_0','xh_1','xh_3','xh_6','xh_10','xh_20'
);
error_0=abs(xh_0(1:3,:)-[xx;yy;zz])';
error_1=abs(xh_1(1:3,:)-[xx;yy;zz])';
error_3=abs(xh_3(1:3,:)-[xx;yy;zz])';
t=1;
for i=1:length(xh_6)
    if isnan(xh_6(1,i))==1
        continue
    end
    error_6(:,t)=abs(xh_6(1:3,i)-[xx(i);yy(i);zz(i)])');
    t=t+1;
end
error_10=abs(xh_10(1:3,:)-[xx;yy;zz])';
redoxh_20=horzcat(xh_20(:,1:768),xh_20(:,770:3891),xh_20(:,3893:end));
redoposition=horzcat(position(1:768,:),position(770:3891,:),position(3893:e
nd,:))';
error_20=abs(horzcat(xh_20(1:3,1:768)-
[xx(1:768);yy(1:768);zz(1:768)],xh_20(1:3,770:3891)-

```

```

[xx(770:3891);yy(770:3891);zz(770:3891)],xh_20(1:3,3893:end)-
[xx(3893:end);yy(3893:end);zz(3893:end)])';
%       +abs(xh_20(1:3,770:3891)-
[xx(770:3891);yy(770:3891);zz(770:3891)])'...
%       +abs(xh_20(1:3,3893:end)-
[xx(3893:end);yy(3893:end);zz(3893:end)])';

mean_rel_error_0=mean(vecnorm((error_0)./[domain(1),domain(2),domain(3)],2,2)
);%[domain(1),domain(2),domain(3)]
mean_rel_error_1=mean(vecnorm((error_1)./[domain(1),domain(2),domain(3)],2,2)
);
mean_rel_error_3=mean(vecnorm((error_3)./[domain(1),domain(2),domain(3)],2,2)
);
mean_rel_error_6=mean(vecnorm((error_6)'./[domain(1),domain(2),domain(3)],2,2)
);
mean_rel_error_10=mean(vecnorm((error_10)./[domain(1),domain(2),domain(3)],2,
2));
mean_rel_error_20=mean(vecnorm((error_20)./[domain(1),domain(2),domain(3)],2,
2));%[0.05,0.05,0.015],2,2));

rotation_error_0=0;
rotation_error_1=0;
rotation_error_3=0;
rotation_error_6=0;
rotation_error_10=0;
rotation_error_20=0;

for i=2:length(xx)
%       position_error=position_error+norm(xh(i,1:3)-[xx(i),yy(i),zz(i)]);

rotation_error_0=rotation_error_0+(xh_0(4:6,i) '*position(i,4:6)')/(norm(posit
ion(i,4:6))*norm(xh_0(4:6,i)));

rotation_error_1=rotation_error_1+(xh_1(4:6,i) '*position(i,4:6)')/(norm(posit
ion(i,4:6))*norm(xh_1(4:6,i)));

rotation_error_3=rotation_error_3+(xh_3(4:6,i) '*position(i,4:6)')/(norm(posit
ion(i,4:6))*norm(xh_3(4:6,i)));

rotation_error_10=rotation_error_10+(xh_10(4:6,i) '*position(i,4:6)')/(norm(po
sition(i,4:6))*norm(xh_10(4:6,i)));
end
for i=2:length(xh_6)
    if isnan(xh_6(1,i))==1
        continue
    end

rotation_error_6=rotation_error_6+(xh_6(4:6,i) '*position(i,4:6)')/(norm(posit
ion(i,4:6))*norm(xh_6(4:6,i)));
end
for i=2:length(redoxh_20)

rotation_error_20=rotation_error_20+(redoxh_20(4:6,i) '*redoposition(4:6,i)')/(
norm(redoposition(4:6,i))*norm(redoxh_20(4:6,i)));
end
% avg_posierror=position_error/length(xx);    %% the unit is m
avg_roterror_0=acosd(rotation_error_0/length(xx));    %% the unit is degree

```

```

avg_roterror_1=acosd(rotation_error_1/length(xx));
avg_roterror_3=acosd(rotation_error_3/length(xx));
avg_roterror_6=acosd(rotation_error_6/3323);%length(xx));
avg_roterror_10=acosd(rotation_error_10/length(xx));
avg_roterror_20=acosd(rotation_error_20/length(redoxh_20));
% error_0_rot=abs(xh_0(:,4:6)*[position(:,4),position(:,5),position(:,6)]');
% error_1_rot=abs(xh_1(:,4:6)-[position(:,4),position(:,5),position(:,6)]);
% error_3_rot=abs(xh_3(:,4:6)-[position(:,4),position(:,5),position(:,6)]);
% error_6_rot=abs(xh_6(:,4:6)-[position(:,4),position(:,5),position(:,6)]);
% error_10_rot=abs(xh_10(:,4:6)-[position(:,4),position(:,5),position(:,6)]);
% error_20_rot=abs(xh_20(:,4:6)-[position(:,4),position(:,5),position(:,6)]);
%
%
mean_rel_error_rot0=mean(vecnorm((error_0_rot)./[domain(4),domain(5),domain(6)
]),2,2));%[domain(1),domain(2),domain(3)]
%
mean_rel_error_rot1=mean(vecnorm((error_1_rot)./[domain(4),domain(5),domain(6)
]),2,2));
%
mean_rel_error_rot3=mean(vecnorm((error_3_rot)./[domain(4),domain(5),domain(6)
]),2,2));
%
mean_rel_error_rot6=mean(vecnorm((error_6_rot)./[domain(4),domain(5),domain(6)
]),2,2));
%
mean_rel_error_rot10=mean(vecnorm((error_10_rot)./[domain(4),domain(5),domain
(6)],2,2));
%
mean_rel_error_rot20=mean(vecnorm((error_20_rot)./[domain(4),domain(5),domain
(6)],2,2));

save('synthetic_PF_newsensor2.mat')

```

8.2.2 Particle Filter Algorithm

```

function [xhk, pf] = particle_filter(initialize,yk, pf, resampling_strategy)
k = pf.k;
if k == 1
    error('error: k must be an integer greater or equal than 2');
end
%% Initialize variables
Ns = pf.Ns; % number of particles
nx = size(pf.particles,1); % number of states
wkml = pf.w(:, k-1); % weights of last iteration
if k == 2
    for i = 1:Ns % simulate initial particles
        pf.particles(:,i,1) = initialize'; % at time k=1
    end
    wkml = repmat(1/Ns, Ns, 1); % all particles have the same weight
end

xkml = pf.particles(:, :, k-1); % extract particles from last iteration;
xk = zeros(size(xkml)); % = zeros(nx,Ns);
wk = zeros(size(wkml)); % = zeros(Ns,1);
sensortable=[0.0545 0 0.0109 0 0 1; 0.0545 0 0.0109 1 0 0;

```

```

0.0545 0 0.0109 0 1 0;    -0.0546 0 0.0109 0 0 1;
-0.0546 0 0.0109 -1 0 0;  -0.0546 0 0.0109 0 -1 0;
0.05435 0.04883 0.0109 0 0 1;    0.05435 0.04883 0.0109 1
0 0;
0.05435 0.04883 0.0109 0 1 0;    -0.04845 0.04772 0.0109
0 0 1;
-0.04845 0.04772 0.0109 -1 0 0;  -0.04845 0.04772 0.0109 0
-1 0 ];
%% Algorithm 3 of Ref [1]
for i = 1:Ns
    % xk(:,i) = sample_vector_from q_xk_given_xkml_yk given xkml(:,i) and yk
    % Using the PRIOR PDF: pf.p_xk_given_xkml: eq 62, Ref 1.

    xk(:,i) = predictstate(xkml(:,i));

    % Equation 48, Ref 1.
    % wk(i) = wkml(i) * p_yk_given_xk(yk, xk(:,i))*p_xk_given_xkml(xk(:,i),
xkml(:,i))/q_xk_given_xkml_yk(xk(:,i), xkml(:,i), yk);

    % weights (when using the PRIOR pdf): eq 63, Ref 1
    % wk(i) = wkml(i) * pf.p_yk_given_xk(yk',
Mag_dipole_pf(xk(:,i)',sensortable)*89)';
    xk_volt=Mag_dipole_pf(xk(:,i)',sensortable)*89;

    S=eye(12).*100;
    wk(i)=exp(-0.5*(xk_volt-yk')'*(S\(xk_volt-yk')));

    % weights (when using the OPTIMAL pdf): eq 53, Ref 1
    % wk(i) = wkml(i) * p_yk_given_xkml(yk, xkml(:,i)); % we do not know this
PDF
end
%% Normalize weight vector
wk= wk./sum(wk);
%% Calculate effective sample size: eq 48, Ref 1
Neff = 1./sum(wk.^2);
resample_percentaje = 0.50;
Nt = resample_percentaje*Ns;

    if Neff < Nt
        disp('Resampling ...')
        [xk, wk] = resample(xk, wk, resampling_strategy);
        % {xk, wk} is an approximate discrete representation of p(x_k | y_{1:k})
    end

%% Compute estimated state
xhk = zeros(nx,1);
for i = 1:Ns
    xhk = xhk + wk(i)'*xk(:,i);
end
%% Store new weights and particles
pf.w(:,k) = wk;
pf.particles(:,:,k) = xk;
return; % bye, bye!!!

%% Resampling function
function [xk, wk, idx] = resample(xk, wk, resampling_strategy)
Ns = length(wk); % Ns = number of particles

```

```

% wk = wk./sum(wk); % normalize weight vector (already done)
switch resampling_strategy
    case 'multinomial_resampling'
        with_replacement = true;
        idx = randsample(1:Ns, Ns, with_replacement, wk);
    %{
        THIS IS EQUIVALENT TO:
        edges = min([0 cumsum(wk)'],1); % protect against accumulated round-off
        edges(end) = 1; % get the upper edge exact
        % this works like the inverse of the empirical distribution and returns
        % the interval where the sample is to be found
        [~, idx] = histc(sort(rand(Ns,1)), edges);
    %}
    case 'systematic_resampling'
        % this is performing latin hypercube sampling on wk
        edges = min([0 cumsum(wk)'],1); % protect against accumulated round-off
        edges(end) = 1; % get the upper edge exact
        u1 = rand/Ns;
        % this works like the inverse of the empirical distribution and returns
        % the interval where the sample is to be found
        [~, idx] = histc(u1:1/Ns:1, edges);
    % case 'regularized_pf' TO BE IMPLEMENTED
    % case 'stratified_sampling' TO BE IMPLEMENTED
    % case 'residual_sampling' TO BE IMPLEMENTED
    otherwise
        error('Resampling strategy not implemented')
end;
xk = xk(:,idx); % extract new particles
wk = repmat(1/Ns, 1, Ns); % now all particles have the same weight
return; % bye, bye!!!

```

8.2.3 Predict State Function

```

function Particlek=predictstate(prevParticles)
% dt=0.001;
prevParticles=prevParticles';

Particlek(1:3) = prevParticles(1:3) + normrnd(0,0.0003, [1 3]); %+ DX1
Particlek(4:6) = prevParticles(4:6) + normrnd(0,0.0005, [1 3]); %Rot*
Particlek(4:6)=Particlek(4:6)/norm(Particlek(4:6))*0.0105;

end

```

8.3 Magnetic Particle Tracking with Sequential Quadratic Programming (SQP) method

8.3.1 Main Code of SQP Algorithm

Note that: SQP algorithm is a fully developed algorithm, so we have utilized this code for many tests at the beginning and this coding part contains algorithm, error analysis, comparison with image-based method etc.

```

clear all; clc;
u0=4*pi*1e-7;

% Convert pixel to real distance
cali=0.1095/norm([2199,1047]-[527,1038]);
zz=0.0109;% zz=10.9mm;
% caliunit1=9.45;
% caliunit2=7.285;
% zz=caliunit2*25;
%Read the chips position from pictures
sensortable=[0.0545  0  0.0109  0  0  1
              0.0545  0  0.0109  1  0  0
              0.0545  0  0.0109  0  1  0
              -0.0546  0  0.0109  0  0  1
              -0.0546  0  0.0109  -1  0  0
              -0.0546  0  0.0109  0  -1  0
              0.05435  0.04883  0.0109  0  0  1
              0.05435  0.04883  0.0109  1  0  0
              0.05435  0.04883  0.0109  0  1  0
              -0.04845  0.04772  0.0109  0  0  1
              -0.04845  0.04772  0.0109  -1  0  0
              -0.04845  0.04772  0.0109  0  -1  0
              ];

chip=zeros;
% ballinitial=[1465,1096,0];
% chip(1,1:3)=[2199,1047,zz]-ballinitial;
% chip(2,1:3)=[527,1038,zz]-ballinitial;

% Convert pixel to real distance
for t=1:4*length(chip)
    chip(t,1:3)=sensortable(3*t,1:3);
    %   chip(t,1:2)=chip(t,1:2)*cali;
    chip(t,2)=-chip(t,2);
    %   chip(t,3)=chip(t,3)/caliunit2;
end

Path='C:\Users\x622t090\Desktop\float_cap3\';
Files1=dir([Path,'rotation_stage.lvm']);
Files2=dir([Path,'nomag1.lvm']);
Files3=dir([Path,'nomag2.lvm']);

fileID2=fopen([Path,'/',Files2.name]);

```



```

% %      %%% REVERSE THE PENDULUM MOTION %%%
% %      %% T=0.9170 seconds l=0.2087 meter
% % %      l=0.2087;
% % %      t=i/100;
% % %      theta0=5/180;
% % %      theta(i)=theta0*cos(sqrt(9.8/l)*t);
% % %      xx(i)=0;
% % %      yy(i)=l*sin(theta(i));
% % %      zz(i)=l*(1-cos(theta(i)));
% end
%
% figure,plot3(xx,yy,zz,'g>-');
% hold on
% plot3(chip(1,1),chip(1,2),chip(1,3),'k*');
% plot3(chip(2,1),chip(2,2),chip(2,3),'k*');
% plot3(chip(3,1),chip(3,2),chip(3,3),'k*');
% plot3(chip(4,1),chip(4,2),chip(4,3),'k*');
% plot3(chip(5,1),chip(5,2),chip(5,3),'k*');
% plot3(chip(6,1),chip(6,2),chip(6,3),'k*');
% plot3(chip(7,1),chip(7,2),chip(7,3),'k*');
% plot3(chip(8,1),chip(8,2),chip(8,3),'k*');
% plot3(chip(9,1),chip(9,2),chip(9,3),'k*');
% plot3(chip(10,1),chip(10,2),chip(10,3),'k*');
% plot3(chip(11,1),chip(11,2),chip(11,3),'k*');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Generate Synthetic Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% for i=1:length(xx)
% %     no_noise=1;
% %     onepercntnoise=(9.9+0.2*rand(1,10))/10;
% for j=1:10
%     r=[chip(j,1)-xx(i), chip(j,2)-yy(i),chip(j,3)-zz(i)];
%     rr=norm(r);
% %     m=[0.0011*sin(pi/20*i),0.0011*sin(pi/20*i),0];
%     a=cos(pi/20*i)*i/(i+log(i+10));
%     b=sin(pi/20*i)*log(i+2)/(1+log(i+30));
%     c=sqrt(1-a^2-b^2);
%     m(i,:)=0.0011*[a,b,c];
% %     m=[ 6.3509e-04, 6.3509e-04, 6.3509e-04];
%     datareading(i,j)=u0/4/pi*...
%         (3*r*(m(i,:)*r')/rr^5-
m(i,:)/rr^3)*Mag(j,:) '*5*10^7;
%
%
% end
% %     datawithnoise(i,:)=awgn(datareading(i,:),10,'measured');
%     datawithnoise(i,:)=datareading(i,:).*(9.9+0.2*rand(1,10))/10;
% end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Check the data disturbance by noise %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fs=2000;
% N = length(datawithnoise(:,1));
% xdft = fft(datawithnoise(:,1));
% xdft = xdft(1:N/2+1);

```

```

% psdx = (1/(Fs*N)) * abs(xdft).^2;
% psdx(2:end-1) = 2*psdx(2:end-1);
% freq = 0:Fs/N:Fs/2;
%
% figure,plot(freq,10*log10(psdx))
% grid on
% title('Periodogram Using FFT')
% xlabel('Frequency (Hz)')
% ylabel('Power/Frequency (dB/Hz)')

% Y=fft(datawithnoise(:,1),1024);
% Pyy = Y.*conj(Y)/1024;
% f = 1000/501*(0:1023);
% figure,plot(f,Pyy(1:1024))
% title('Power spectral density')
% xlabel('Frequency (Hz)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter of high frequency noise %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fs = 500;
% Hd = designfilt('lowpassfir','FilterOrder',20,'CutoffFrequency',200, ...
%     'DesignMethod','window','Window',{'kaiser',3},'SampleRate',Fs);
% % datawithnoise(1:2000,:)=[];
% y1 = filter(Hd,datawithnoise(:,1));
%
%
figure,plot(1:length(datawithnoise),datawithnoise(:,1),1:length(datawithnoise),y1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*****%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dipole_position=[xx;yy;zz]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Chip position calibration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X(1,:)= [0,0,0,0.0010,0.0003,0];
% X(1:10,1:3)=chip*0.80;
% for m=1:10
% X(m,4:6)=[6.3509e-04, 6.3509e-04, 6.3509e-04];
% end
% for chipth=1:7%length(chip)
%     options = optimoptions('fmincon','Algorithm','sqp');
%
gs2=GlobalSearch('Display','off','NumStageOnePoints',500,'NumTrialPoints',2e3,
'StartPointsToRun','bounds-ineqs');
%
%
f=@(x) chipcali(x,datawithnoise(:,chipth),dipole_position(:,1:3),chipth);
%
%     problem2 =
createOptimProblem('fmincon','objective',f,'x0',X(chipth,:), 'lb', [-0.03,-
0.020,0,-0.0011,-0.0011,-
0.0011], 'ub', [0.03,0.020,0.04,0.0011,0.0011,0.0011], 'nonlcon',@my_nlincon, 'options',options);
%
[chip_position1(chipth,:),fvalchip(chipth),exitflag(chipth),output]=run(gs2,problem2);

```

```

% %         if fvalchip(chipth)>5
% %             continue
% %         end
%         chips(chipth,:)=chip_position1(chipth,:);
% end
%
% save('chips.mat','chips');
% load('chips.mat','chips');

% X(1,:)= [0,0,0.05,0.0010,0.0003,0];

X(1,:)= [0,0,0.02,0.010,0.001,0];
% UB=[max(xx)+0.02,max(yy)+0.02,max(zz)+0.02,0.0011,0.0011,0.0011];
% LB=[min(xx)-0.02,min(yy)-0.02,min(zz)-0.02,-0.0011,-0.0011,-0.0011];

LB=[-0.04,0.01,0.01,-0.0105,-0.0105,-0.0105];
UB=[0.047,0.03,0.06,0.0105,0.0105,0.0105];
% tic
options = optimoptions('fmincon','Algorithm','sqp');
f=@(x)Quality1_direct(x,datawithnoise(1,:),chip(:,1:3));
problem1 =
createOptimProblem('fmincon','objective',f,'x0',X(1,:), 'lb',LB, 'ub',UB, 'nonlc
on',@my_nlincon, 'options', options);

gs=GlobalSearch('Display','off','NumStageOnePoints',500,'NumTrialPoints',3e3,
'StartPointsToRun','bounds-ineqs');

gs2=GlobalSearch('Display','off','NumStageOnePoints',500,'NumTrialPoints',3e3
,'StartPointsToRun','bounds-ineqs')
[position1(1,:),fvalinit,exitflag(1),output]=run(gs,problem1);
X(2,:)=position1(1,:);
globalfindnum=1;
bigjump=1;
for j=2:length(datawithnoise)%200%
%     timestep=j;
%     if max(abs(datawithnoise(j,:)))>2000
%         continue
%     end
%     LB=[position1(1,1)-0.002,position1(1,2)-0.002,position1(1,3)-
0.002,position1(1,4)-0.0002,position1(1,5)-0.002,position1(1,6)-0.002];
%
UB=[position1(1,1)+0.002,position1(1,2)+0.002,position1(1,3)+0.002,position1(
1,4)+0.0002,position1(1,5)+0.002,position1(1,6)+0.002];
f=@(x)Quality1_direct(x,datawithnoise(j,:),chip(:,1:3));
%
% problem =
createOptimProblem('fmincon','objective',f,'x0',X(j,:), 'lb', [], 'ub', [], 'nonlc
on',@my_nlincon, 'options', options);
% problem = createOptimProblem('fmincon','objective',f,'x0',X(j-
2,:), 'lb', [], 'ub', [], 'options', options);
problem2 =
createOptimProblem('fmincon','objective',f,'x0',X(j,:), 'lb',LB, 'ub',UB, 'nonlc
on',@my_nlincon, 'options', options);
% gs2=GlobalSearch;
%
gs=GlobalSearch('Display','off','NumStageOnePoints',200,'StartPointsToRun','b
ounds-ineqs')

```

```

% position1(1,:)=run(gs,problem);

[position1(j,:),fval(j),exitflag(j),output]=fmincon(problem2);
act=0;
if exitflag(j)<2 | fval(j)>0.008
%   act=act+1;
   globalfindnum=globalfindnum+1;
   [position1(j,:),fval(j),exitflag(j),output]=run(gs2,problem2);
end
if norm(position1(j,1:3)-position1(j-1,1:3))>0.0025
   local_LB=position1(j-1,:)-[0.001 0.001 0.001 0.001 0.001 0.001];
   local_UB=position1(j-1,:)+[0.001 0.001 0.001 0.001 0.001 0.001];
   problem3 =
createOptimProblem('fmincon','objective',f,'x0',X(j,:), 'lb',local_LB,'ub',local_UB,'nonlcon',@my_nlincon,'options',options);

   [position1(j,:),fval(j),exitflag(j),output]=run(gs2,problem3);
   bigjump(j)=1;
end
X(j+1,:)=position1(j,:);
end
% save('float_cap.mat')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Image starts at 140 ends at 4386
figure,plot(position1(1567:2:end,1),position1(1567:2:end,2),'-*');axis equal
figure,plot(-ballposi2(1:5:end,1),ballposi2(1:5:end,2));axis equal

figure,plot(position1(3:2:end,1),'-*');
hold on
plot(-ballposi2(1:5:end,1)+0.0675);
% plot(-10*magballx2(618+74*5/2:5:end)+0.05);

figure,plot(position1(3:2:end,2),'*--')
hold on
plot(ballposi2(1:5:end,2)-0.0793);
% plot(-10*magbally2(618+5*74/2:5:end)+0.072);

% save('floatcap4_sqp_withbigjump.mat')
% cali=4940;
% magballx2=-magballx/cali;
% magbally2=-magbally/cali;
plot(10*magballx2(425:5:end),10*magbally2(425:5:end),'s');

figure,plot(position1(821:2:6762,1),'-*'); %100 Hz
hold on
plot(-10*magballx2(425:5:end)+0.05,'-'); %100 Hz

figure,plot(position1(821:2:6762,2),'-*'); %100 Hz
hold on
plot(-10*magbally2(425:5:end)+0.076,'-'); %100 Hz

% save('floatcap2_sqp.mat')

% cam2pix=0.051/335;
% center2=center*cam2pix;

```

```

% xout2=xout.*cam2pix-0.0253;
% yout2=yout.*cam2pix-0.002;
% xin2=xin.*cam2pix-0.0253;
% yin2=yin.*cam2pix-0.002;
% center2(:,:)= [center2(:,1)-0.0253,center2(:,2)-0.002];
%
% figure,plot(position1(1:10:end,1),position1(1:10:end,2),'*-');
% hold on
% plot(position1(1:5:15,1),position1(1:5:15,2),'r')
% plot(center2(:,1),center2(:,2),'s');
% % plot(xout2,yout2);
% % plot(xin2,yin2);
% axis equal
% xlabel('X, meter','FontSize',12,'FontWeight','bold');
% ylabel('Z, meter','FontSize',12,'FontWeight','bold');
% % title('Camera result vs Reconstructed trajectory in x z plane');
% legend('SQP reconstruct','camera
label','outline','innerline','FontSize',10,'FontWeight','bold');
%
% saveas(gcf,['C:\Users\x622t090\Desktop\MPT figures\','Circle_sqp','.png']);

%
% figure,plot(Recons_position(50:5:end,1),Recons_position(50:5:end,3),'*-');
% hold on
% plot(center2(:,1),center2(:,2),'s');
% % plot(xout2,yout2);
% % plot(xin2,yin2);
% axis equal
% xlabel('X, meter','FontSize',12,'FontWeight','bold');
% ylabel('Z, meter','FontSize',12,'FontWeight','bold');
% % title('Camera result vs Reconstructed trajectory in x z plane');
% legend('EKF reconstruct','camera
label','outline','innerline','FontSize',10,'FontWeight','bold');
%
% saveas(gcf,['C:\Users\x622t090\Desktop\MPT figures\','Circle_ekf','.png']);
%
%
% LL=length(Recons_position(1:50:end,4));
%
figure,quiver3(zeros(1,LL)',zeros(1,LL)',zeros(1,LL)',Recons_position(1:50:en
d,4),Recons_position(1:50:end,5),Recons_position(1:50:end,6))
% xlabel('Mx, Tesla','FontSize',12,'FontWeight','bold');
% ylabel('My, Tesla','FontSize',12,'FontWeight','bold');
% zlabel('Mz, Tesla','FontSize',12,'FontWeight','bold');
% saveas(gcf,['C:\Users\x622t090\Desktop\MPT
figures\','Magnet_moment','.png']);
%
%
% figure,plot(position1(:,1));
% hold on
% plot(position1(:,2));
% plot(position1(:,3));
% xlabel('timestep');
% ylabel('position')
% legend('x','y','z')
% saveas(gcf,['C:\Users\x622t090\Desktop\MPT
figures\','Circle_xyz_sqp','.png']);

```

```

%
%
% figure,plot(position1(:,4));
% hold on
% plot(position1(:,5));
% plot(position1(:,6));
% % figure,plot(center(:,1));
% xlabel('timestep','FontSize',12,'FontWeight','bold');
% ylabel('magnet strength, Tesla','FontSize',12,'FontWeight','bold')
% legend('Mx','My','Mz')
% title('Magnet strength change with time')
% saveas(gcf,['C:\Users\x622t090\Desktop\MPT
figures\'','Mx_My_Mz_sqp','.png']);

% plot(position1(1:1:3,1),position1(1:1:3,2),'*-');
% plot3(xx,yy,zz,'gs-');
% axis equal
% xlabel('m','FontSize',16,'FontWeight','bold','Color','k')
% ylabel('m','FontSize',16,'FontWeight','bold','Color','k')
% zlabel('m','FontSize',16,'FontWeight','bold','Color','k')
% set(gca,'FontSize',16);
% legend({'reconstruct trajectory','arbitrary path'],'Position',[0.8 0.85
0.005 0.001],'FontSize',10); %'grid=1.5','
% figure,plot(position1(:,4),'-^');
% hold on
% plot(m(:,1),'*-')

%%%%%%%%%%%%%% Error Estimation %%%%%%%%%%%%%%%
% position_error=0;
% rotation_error=0;
% for i=1:length(xx)
%     position_error=position_error+norm(position1(i,1:3)-
[xx(i),yy(i),zz(i)]);
%
rotation_error=rotation_error+(position1(i,4:6)*m(i,:))/(norm(m(i,:))*norm(p
osition1(i,4:6)));
% end
% avg_posierror=position_error/length(xx); %% the unit is m
% avg_roterror=acosd(rotation_error/length(xx)); %% the unit is degree
%
% FFA=figure,plot(position1(80:220,1),'*')
% hold on
% plot(xx(80:220),'-bs')
% plot(position1(80:220,2),'*')
% plot(yy(80:220),'-rs')
% plot(position1(80:220,3),'*')
% plot(zz(80:220),'-ys')
% title('Error of x y z coordinate')
% xlabel('timestep','FontSize',16,'FontWeight','bold','Color','k')
% ylabel('mm','FontSize',16,'FontWeight','bold','Color','k')
% set(gca,'FontSize',16);
% legend({'reconstruct x coordinate','arbitrary path x coordinate',...
'reconstruct y coordinate','arbitrary path y coordinate',...
'reconstruct z coordinate','arbitrary path z coordinate'},...
'FontSize',12,'Location','southeast'); %'grid=1.5','position',[120 -
0.01,0.02,0.01],
%
%
```

```

% % str=num2str()
% save(FFA,'C:\Users\x622t090\Desktop\MPT figures\no_noisexyz.jpg');
% save('newdaqpend1.mat')

% save('Newsensor_sqp_circle1.mat')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Velocity and Angular Velocity %%%%%%%%%%
% vx=Differentialkernel(position1(:,1));
% vy=Differentialkernel(position1(:,2));
% vz=Differentialkernel(position1(:,3));
%
% for i=1:length(position1)-1
% % angular=
[position1(i+1,4),position1(i+1,5),position1(i+1,6)]*[position1(i,4),position
1(i,5),position1(i,6)]'/.
% %
(norm([position1(i+1,4),position1(i+1,5),position1(i+1,6)])*norm([position1(i
,4),position1(i,5),position1(i,6)]));
% angvel(i,:)=([position1(i+1,4),position1(i+1,5),position1(i+1,6)]-
[position1(i,4),position1(i,5),position1(i,6)]);
% end

```

8.3.2 The quality function to be minimized

```

function zzz=Qquality1(x,datawithnoise,chip)%
% load('ringpath.mat')
% ringpath=ringpath(1:50:end,:);
u0=4*pi*1e-7;
Mag(1,:)= [0 0 1];

Mag(2,:)= [1 0 0];

Mag(3,:)= [0 1 0];

Mag(4,:)= [0 0 1];

Mag(5,:)= [-1 0 0];

Mag(6,:)= [0 -1 0];

Mag(7,:)= [0 0 1];

Mag(8,:)= [1 0 0];

Mag(9,:)= [0 1 0];

Mag(10,:)= [0 0 1];

Mag(11,:)= [-1 0 0];

Mag(12,:)= [0 -1 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

sum=0; act=0;
weight=zeros;
for datanum=1:12%length(datawithnoise) %[3,4,5,7,8]%

    if rem(datanum,3)==0
        chipsnum=datanum/3;
    else
        chipsnum=fix(datanum/3)+1;
    end

%     if abs(datawithnoise(datanum))>1900 | abs(datawithnoise(datanum))<10
%         continue
%     end
%     if act>6
%         break
%     end
if abs(datawithnoise(datanum))<=15
    weight(datanum)=abs(datawithnoise(datanum))/15;
elseif abs(datawithnoise(datanum))>15 && abs(datawithnoise(datanum))<1900
    weight(datanum)=1;
else abs(datawithnoise(datanum))>=1900
    weight(datanum)=0.001;
end
%     R=[chip(chipsnum,1)-x(1), chip(chipsnum,2)-x(2), chip(chipsnum,3)-
x(3)];
%     rr=norm(R);
%     m=[x(4),x(5),x(6)];
%     B=u0/4/pi*10^6*(3*R*m'*R/rr^2-m);
%     sum=sum+abs(B* Mag(datanum,:)'-datawithnoise(chipsnum)*rr^3/89);
sum=sum+weight(datanum)*abs( ...
    1e6*u0/4/pi*(...
        3*(...
            [chip(chipsnum,1)-x(1), chip(chipsnum,2)-x(2),
chip(chipsnum,3)-x(3)] ...
                *( [x(4),x(5),x(6)] * [chip(chipsnum,1)-x(1), chip(chipsnum,2)-
x(2), chip(chipsnum,3)-x(3)]' ) ...
                    ) / (norm([chip(chipsnum,1)-x(1), chip(chipsnum,2)-
x(2), chip(chipsnum,3)-x(3)])^2 - [x(4),x(5),x(6)] ...
                        ) * Mag(datanum,:)') ...
        - (datawithnoise(datanum)/89) ...
        * (norm([chip(chipsnum,1)-x(1), chip(chipsnum,2)-
x(2), chip(chipsnum,3)-x(3)])^3 ...
            )
        );
end
%     R=[chip(chipsnum,1)-x(1), chip(chipsnum,2)-x(2), chip(chipsnum,3)-
x(3)];
%     rr=norm(R);
%     m=[x(4),x(5),x(6)];
%     B=u0/4/pi*(3*R*m'*R/rr^2-m)/rr^3*89;
%     zzz=zzz+abs(B*chip(chipsnum,4:6)'-datawithnoise(chipsnum)*rr^3/89);
zzz=sum;

```