RASPBERRY PI-BASED SENSOR NETWORK FOR MULTI-PURPOSE NONLINEAR MOTION DETECTION IN LABORATORIES USING MEMS

RED DE SENSORES MULTIFUNCIONAL BASADA EN RASPBERRY PI PARA LA DETECCIÓN DE MOVIMIENTO NO LINEAL EN LABORATORIOS USANDO MEMS

Natarajan Shriethar[1], **Narmadha Chandramohan**[2], **Chandramohan Rathinam**[3]

[1] Department of Energy Science, Alagappa University, Karaikudi. India.
[2] Vigil Health Solutions, Canada.
[3] Vidyaagiri Coleege of Arts and Science, Puduvayal, India.

(Recibido: 05/2022. Aceptado: 06/2022)

Abstract

Detecting and measuring the nonlinear evolution of a classical system is a complex process. In this work, the nonlinear evolution of a lab-level system is measured using the Raspberry Pi-based MEMS sensor network system. The evolution and various results of nonlinear systems were compared theoretical and experimentally.

Keywords: motion sensors, Lagrangean solution, Raspberry Pi, nonlinear measurement system, MEMS.

Resumen

Detectar y medir la evolución no lineal de un sistema clásico es un proceso complejo. En este trabajo, se mide la evolución no lineal de un sistema a nivel de laboratorio utilizando el sistema de red de sensores MEMS basado en Raspberry Pi. La evolución y varios resultados de sistemas no lineales fueron comparados teórica y experimentalmente.

Palabras clave: sensores de movimiento, solución Lagrange, Raspberry Pi, sistema de medición no lineal, MEMS.

N. Shriethar: natarajangravity@gmail.com https://doi.org/10.15446/mo.n65.102641

Introduction

Real life exhibits many nonlinear systems. Understanding the dynamics of nonlinear systems requires complex algorithms and equations, but from the computational perspective, the simple and elegant Raspberry Pi machine [1] solves these problems by analyzing the nonlinear evolution in real time. Raspberry Pi has many applications in modern days.

Raspberry Pi are applied to agricultural applications [2], Healthcare applications [3, 4], biotechnology applications [5], motion contol sytems sytems [6], deep learning systems [7] and much more.

When using the Raspberry Pi, thermal measurements are taken and their data are recorded on a server [8] and, by using its thermal sensor networks, a solar thermal laboratory is automated [9]. Via these pins, the real world can be communicated to the computer. Micro-electromechanical systems technology is shortly referred to as MEMS [10]. The MEMS devices can have physical sizes from well below one micron to a few millimeters. The basic elements of MEMS are embedded microsensors, actuators, and other microelectronics. Microsensors convert the mechanical signal into an electrical signal. Physical parameters such as temperature, pressure, inertial forces, chemical species, magnetic fields and radiation are observed through MEMS devices. From the atomic or molecular level, nanotechnology helps to produce MEMS devices at the nano-dimensional scale. The MPU9250 MEMS sensor is applied to unmanned aerial vehicle controls [11], real-time dynamics reorganization [12], altitude estimation [13], pedestrian tracking [14], self-balancing robot design [15] and many more.

The aim of this work is to measure the nonlinear dynamics of a system using a Raspberry Pi-based sensor setup. To achieve this, an IMU sensor is needed. IMU is referred to as Inertial Measurement Units. These are sensors applied to understand the dynamics of a body in three-dimensional space. IMU sensors are used to measure roll- rotation along X-axis ϕ, Pitch - rotation along Y-axis θ, and Yaw - rotation along Z-axis ψ.

These IMU sensors can provide information about the physical body's orientation, angular rate, and force applied to it by measuring the linear and angular motion using a combination of a gyroscope, accelerometer, and magnetometer. In the IMU sensor, the gyroscope module measures the rate of rotation, as well as tracking these rotations over time. Overall, this module calculates the current angle. A gyroscope module is implemented for measuring fast and sharp movements. In general, the Inertial Measurement Units are implemented as a combination of accelerometer and gyroscope. The real-time nonlinear system from the laboratory instrument is considered. The data is pinged to the Raspberry Pi and the analysis is done. For observing nonlinearity, the Lagrangean solutions are obtained.

Tools

Raspberry Pi is a single-board computer that has been recently used in various fields. The Raspberry Pi comes in several variants and the most notable ones are Rasberry Pi 4 Model B, Raspberry Pi Pico, Raspberry Pi Model B, and Raspberry Pi Zero W. These various versions of Raspberry Pi have affordable prices, such as the Raspberry Pi Pico, which costs approximately only 4 USD.

The Raspberry Pi 4 Model B has advanced configurations in addition to all other variants. It features Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model) 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless chipsets, and Bluetooth 5.0, BLE Gigabit Ethernet. For connectivity, the Raspberry Pi 4 has 2 USB 3.0 ports and 2 USB 2.0 ports. GPiO (General Purpose Input Output) pins are the exceptional peripheral of Raspberry Pi. Raspberry Pi 4 has a standard 40 pin GPiO header. For the display and audio-video, the Raspberry Pi 4 has 2 micro-HDMI ports, 2-lane MIPi DSI display port, 2-lane MIPi CSI camera port, 4-pole stereo audio and composite video port. It comes with OpenGL ES 3.1 and Vulkan 1.0 licenses.

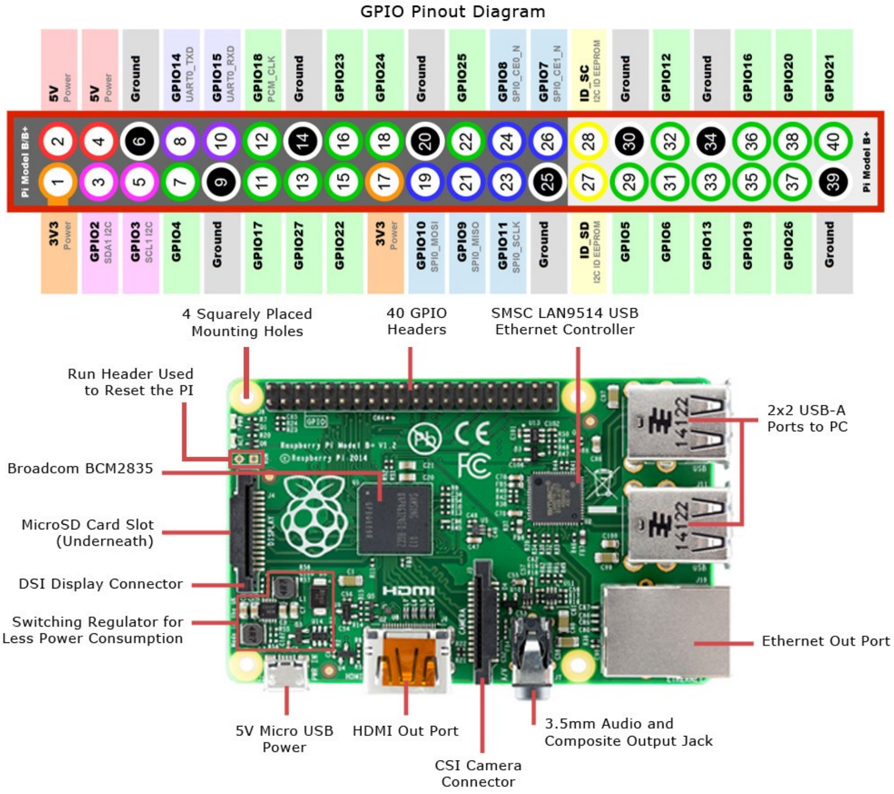

FIGURE 1. Raspberry Pi and GPiO.

The operating system and other data are loaded into the micro SD card in the Raspberry Pi. Hence, it has an onboard micro SD card slot for loading the operating system and data storage. In addition to these, it also has 5V DC via USB-C connector (minimum 3A*), 5V DC via GPIO header (minimum 3A*), and an Ethernet port. The Raspberry Pi has an operating temperature of 0 to 50 degrees Celcius.

For the simpler version of the nonlinear sensing and relatively smaller experiments, the Raspberry Pi Pico is considered. The Raspberry Pi Pico has the following configurations: dual-core Arm Cortex-M0+ processor with 264KB internal RAM and it can support up to 16MB of off-chip memory. It has other salient features like I2C, SPI, and uniquely programmable I/O (PiO).

The Raspberry Pi foundation aims to provide computing for everyone and focuses on research areas such as Gender Balance in Computing program, Culturally responsive teaching, Computing education and learners from low-income families, Non-programming aspects of the curriculum, Classroom talk in programming, Computer science for learners aged 14 to 18, AI and data science education, AstroPi and more.

The MPU9250 is a 9 axis sensor. The sensor is calibrated for the local domain and it is connected to the Raspberry Pi for the data logging and sensing. The MPU-9250 is a 9-axis (gyro + accelerometer + compass) IMU. It is a MEMS motion tracking device. It has I2C and SPI Interface Types. Its operating supply current is 3.7 mA and its operating supply voltage is 2.5 V. The minimum operating temperature of the MPU9250 is -40 °C, and its maximum operating temperature is +85 °C. It weighs as low as 110 mg. The MPU9250 has a triple-axis MEMS. The range of the accelerometer in full-scale is $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$. The ranges of the gyroscope are, ±250, ±500, ±1000, and ±2000°/sec. The full-scale range of the magnetometer is $\pm 4800\mu T$.

The gyroscope operating current is 3.2 μA; the accelerometer normal operating current is 450 μA; the magnetometer normal operating current is 280 μA, and the sleep mode operating current is 8 μA. Accelerometers are applied on detecting gravity-based static movements and dynamic movements based on inertia and acceleration. A magnetometer module uses the earth's magnetic field to understand the direction of the sensor and is built based on the Hall effect.

The experimental setup is designed with the following peripherals: the real-time instrument with nonlinear dynamics is attached to the IMU sensor MPU9250. The sensor is further attached with the Raspberry Pi for data processing. The sensor and the Raspberry Pi are powered with a dedicated portable power supply. This dedicated power supply is for the wireless data access of the nonlinear setup. The Raspberry Pi can be accessed headlessly.

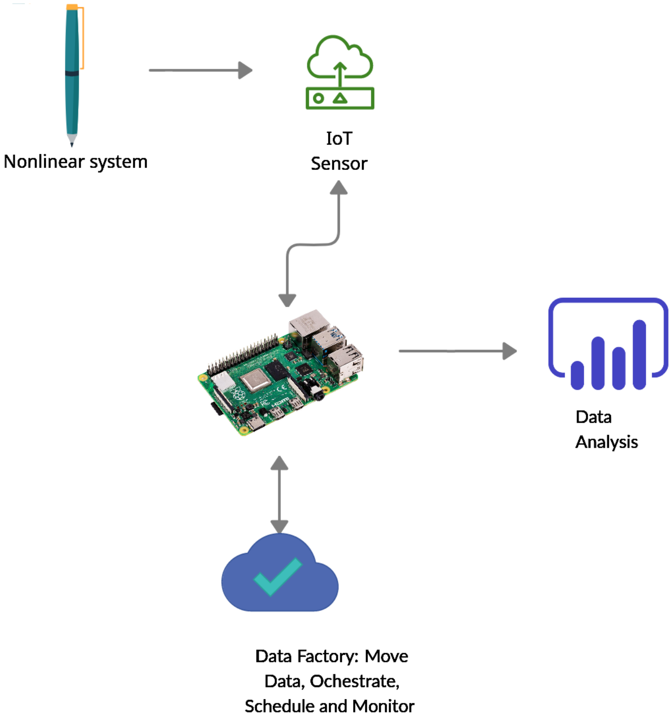

FIGURE 2. RaspberryPi Nonlinear sensing setup.

Hence, the physical interference in the setup is minimized. The Raspberry Pi is connected to the cloud server too, so the seamless flow of data is also pinged to the cloud. That leads to the distribution of the data over any network nodes via the cloud.

As the corresponding Python code is executed, the MPU9250 starts to sense the real-time variations of the gyroscope, accelerometer, and magnetometer. The detected data are immediately pinged to the Raspberry Pi. Those data sets are stored on the local server for analysis. Similarly, those data are served to the dedicative cloud simultaneously.

The stored data are subsequently analyzed using data analysis tools. Through data analysis, various information about the nonlinear instrument is understood.

Simple python program

Python is a very convenient language for the Raspberry Pi-based computations [16, 17]. The combo of python and Raspberry Pi is even used in the fields of game development [18]. For nonlinear sensing using Raspberry Pi, python program is implemented to sense and log the data. The Python program obtains the sensor data and logs it to the Raspberry Pi as well as to the cloud server [19]. Once the *FaBo9AxisMPU9250* module is installed via *pip*, the following python code can be executed to obtain the sensed values from the MPU9250.

```
import FaBo9Axis_MPU9250
import time
import sys

MPU9250 = FaBo9Axis_MPU9250.MPU9250()

try:
while True:
accel = MPU9250.readAccel()
print " ax = " , ( accel['x'] )
print " ay = " , ( accel['y'] )
print " az = " , ( accel['z'] )

gyro = MPU9250.readGyro()
print " gx = " , ( gyro['x'] )
print " gy = " , ( gyro['y'] )
print " gz = " , ( gyro['z'] )

mag = MPU9250.readMagnet()
print " mx = " , ( mag['x'] )
print " my = " , ( mag['y'] )
print " mz = " , ( mag['z'] )
print

time.sleep(0.1)

except KeyboardInterrupt:
sys.exit()
```

The python code prints the 3-dimensional values from the accelerometer, gyroscope, and magnetometer. The continuous observational sensed data is printed in the terminal window as the output of the python program. The program is written based on the Menegazzo *et al.* work [19].

Experiment

The MPU9250 sensor is attached to the Raspberry Pi and the data is communicated via I2C protocol. The I2C protocol has to be enabled manually before setting the IMU sensors. The Raspberry Pi Pins that have the mapPing with MPU9250 are Pin 1 as VCC, Pin 6 is GND, Pin 5 as SCL, and Pin 3 as SDA. Whereas SCL and SDA are the major peripherals of the MPU9250 sensor, in MPU9250 SCL is an SPi serial port clock. It behaves as an I2C serial clock. It usually exhibits 100 or 400 kHz frequency in I2C and up to 1 MHz in SPi. In the same sensor, SDA refers I2C serial data hub, and it can also be used for SPi serial data input (SDI). It has a voltage input pin which is VDD. Usually, its power supply is in the range of +2.4V to +3.6V, and GND indicates the ground reference pin.

Before the observation is done, the calibration of the sensors is required. According to the laboratory environment, it is very important to calibrate those sensors. The experiment is designed to observe the hand movements of the performer via the Raspberry Pi device and sensor networks. The MPU9250 sensor is a motion detection device that can read the object's motion in 360 degrees. Therefore, the Raspberry Pi device with an attached sensor configuration is attached to the front of the performer. For live data analysis, the detected data are sent to a server in the cloud. The entire configuration is packaged in a small volume for the convenience of the performer. For analysis, an open source tool QtiPlot is implemented and, for calculation, the Python programming language is used.

Solution

The Lagrangean solution is obtained for classical systems which move under certain constraints [20]. For the nonlinear experimental setup, the Lagrangian of the two-arm dynamics is considered. As the system faces a nonlinear evolution, and subsequently turns out to be a chaotic evolution, the Lagrangian solutions for this experiment are obtained.

From the Lagrangean perspective, the force is derived for single hand nonlinear evolution.

$$F(r) = -gl_0m_1\cos(q_1(t)) \tag{1}$$

In equation 1, q refers to the generalised coordinated system. In plots, u denoted generalized velocities, m indicates the mass of the objects at the end of the arm, l refers to the length of each link, g and t refer to the acceleration of gravity and time.

For the double leg system, the force is derived in terms of Lagrangean as

$$F(r) = -gl_0m_1\cos(q_1(t)) - -gl_0m_2\cos(q_1(t)) - -gl_1m_2\cos(q_2(t)) \tag{2}$$

The Lagrangean for small nonlinear oscillations is obtained as

$$L = ma^2[\dot{\theta}^2(1+2\sin^2\theta) + \Omega^2\sin^2\theta + 2\Omega_0\cos\theta] \tag{3}$$

If that system is nonlinear double pendulum, then the Lagrangean is obtained as

$$\begin{aligned} L = \frac{1}{2}(m_1+m_2)l_1^2\dot{\theta}_1^2 &+ \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2) \\ &+ (m_1+m_2)gl_1\cos\theta_1 + m_2gl_2\cos\theta_2 \end{aligned} \tag{4}$$

Discussion

In figure 3, the theroretical analysis of nonlinerar dyanamics of two arm instrumentation is plotted. The derivation from 4 offerst the data for the given plot. The rendomness is obtained for various veloocities of the various arms of the pendulum. The pure randomness is shown for the various time intervals as various velocities.

For simple nonlinear system, the graph is ploted for the theoretical predictions obtained in 3. The graph 4 shows the theoretical variations in the acceleration data with randomness. But this randomness is relatively low as compared to the graph 3. This lower randomness is obtained as the non linear system has single arms.

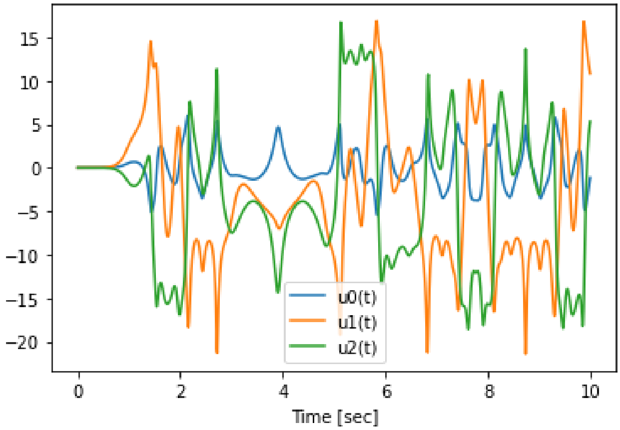

Figure 3. *Theoretical curve.*

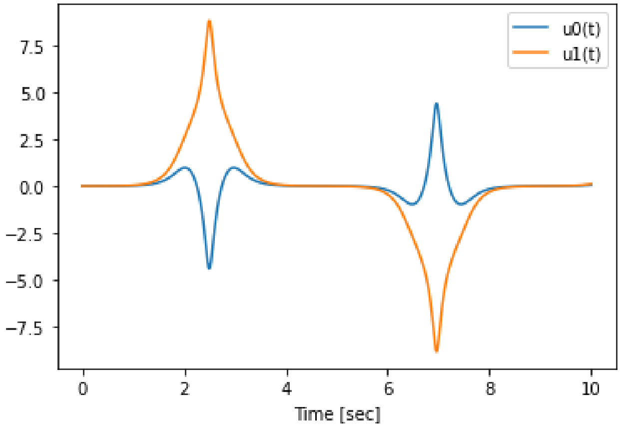

Figure 4. *Theoretical single hand curve.*

In figure 5, the experimental data of the nonlinear instrument is plotted from the output data of MPU9250. The nonlinearity is obtained due to the nonlinear motion of the arms of the instrument. The respective instrument changes its arm on a random timescale. The acceleration data is plotted for the variations. The randomness exists in various places in the graph. The graph shows sudden increments and decrements in the peaks of the velocity data. Also, after 2.4×10^{-5} seconds, there exist a rapid hike in the velocity, and around 2.6×10^{-5} there exists a rapid fall down from the peaked data. The highs and lows in the

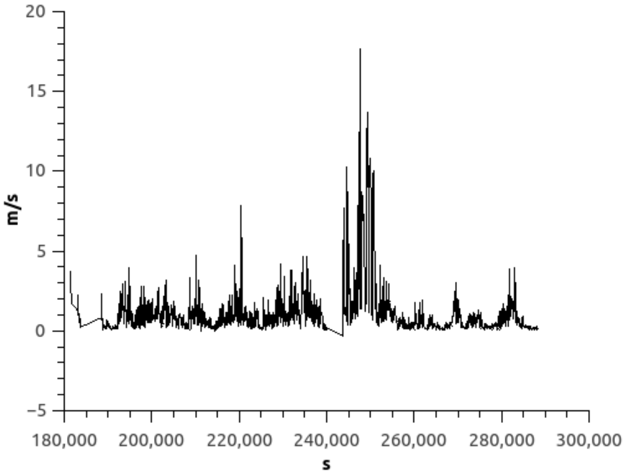

FIGURE 5. *Experimental curve.*

data exhibit the rapid movement of the arms of the instrument in a nonlinear fashion. This confirms the theoretical predictions obtained in figures 4 and 3. Hence, the longevity of the sensor is also confirmed in this comparison, so the Raspberry Pi-based IMU sensor networks can be implemented in the laboratories for nonlinear measurements.

Conclusion

The Raspberry Pi-based IMU sensors are the very right tools to analyze the dynamics in nonlinear labs. This setup can be implemented in various nonlinear labs such as thin-film labs, crystal growth labs, photovoltaic labs, biochemical labs, and other nonlinear dynamical labs. These labs can utilize these setups to understand and design their types of equipment and instruments. Once proper calibrations are achieved, the sensor exhibit very good output for long periods. In addition, the entire configuration is equipped with low voltage, and the small volume makes this configuration suitable for installation in any laboratory. For simplified and lightweight experiments, the variants of Raspberry Pi such as Raspberry Pi Pico and Raspberry Pi Zero can also be used. Additionally, as the Raspberry Pi interfacing can be achieved with headless mode, the setup also comes under low-cost devices. The prescribed experiment senses and logs the nonlinear data from the real world using MPU9250 sensors. This similar setup can be extended to any other dynamical system such as real-time motion of a physical body, detection of human body movements, sports analysis, and much more.

References

[1] E. Upton and G. Halfacree, *Meet the raspberry Pi* (John Wiley & Sons, 2012).

[2] S. E. Mathe, M. Bandaru, H. K. Kondaveeti, S. Vappangi, and G. S. Rao, in *2022 International Conference on Innovative Trends in Information Technology (ICITIIT)* (IEEE, 2022) pp. 1–7.

[3] S. Sengan, O. I. Khalaf, S. Priyadarsini, D. K. Sharma, K. Amarendra, and A. A. Hamad, Int. J. of Reliable and Quality E-Healthcare (IJRQEH) **11**, 1 (2022).

[4] S. Naik and E. Sudarshan, ARPN J. Eng. Appl. Sci. **14**, 872 (2019).

[5] S. Vappangi, N. K. Penjarla, S. E. Mathe, and H. K. Kondaveeti, in *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)* (IEEE, 2022) pp. 1–6.

[6] Z. Didi, I. El Azami, and E. M. Boumait, in *International Conference on Digital Technologies and Applications* (Springer, 2022) pp. 427–434.

[7] D. K. Dewangan and S. P. Sahu, IEEE sensors J. **21**, 3570 (2020).

[8] S. Natarajan, A. Deepika, I. Pradeeba, and R. Chandramohan, Int Research J Eng and Technol, Trichy **4**, 254 (2017).

[9] N. Shriethar, P. Letchoumanane, and S. Solai, Momento , 1 (2022).

[10] M. Gad-el Hak, *The MEMS handbook* (CRC press, 2001).

[11] Y. Tang and S. Lin, in *IOP Conference Series: Earth and Environmental Science*, Vol. 621 (IOP Publishing, 2021) p. 012151.

[12] S. Bommannan, C. Vineeth, U. Mylavarapu, S. Boyanapalli, and S. Vidhya, in *Proceedings of International Conference on Intelligent Computing, Information and Control Systems* (Springer, 2021) pp. 39–55.

[13] Q. Zhao, Y. Fu, Z. Liu, Y. Xu, and X. Liu, in *Automatic Control, Mechatronics and Industrial Engineering: Proceedings of the International Conference on Automatic Control, Mechatronics and Industrial Engineering (ACMIE 2018), October 29-31, 2018, Suzhou, China* (CRC Press, 2019) p. 197.

[14] J. Zheng, Q. Minhui, K. Xiang, and M. Pang, in *International Conference on Intelligent Robotics and Applications* (Springer, 2017) pp. 494–504.

[15] Y. Zhang and W. Cai, in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (IEEE, 2021) pp. 1039–1043.

[16] W. Donat and C. Krause, *Learn Raspberry Pi Programming with Python* (Springer, 2014).

[17] T. Cox, *Raspberry Pi Cookbook for python programmers* (Packt Publishing Ltd, 2014).

[18] S. Kelly, *Python, PyGame, and Raspberry Pi Game Development* (Springer, 2019).

[19] J. Menegazzo and A. Wangenheim, "MPU-9250 Sensors Data Collect," (2020).

[20] H. A. Kastrup, Physics Reports **101**, 1 (1983).