

Szymon Firlej\*

## **Design, construction and control of a spherical rolling robot with internal two-wheel cart**

### **1. Introduction**

The extreme maneuverability of ball-shaped objects is remarkable, thus making it worthy to pursue the development of spherical robot designs. Another important advantage of such robots is their resistance to hostile environment conditions.

A study of the literature has shown that there are at least four different propulsion types available to use in spherical robot designs. Propulsion based on the relocation of a robot's center of mass is the most-common solution. This is sometimes paired with the flywheel boost mechanism, offering additional torque in order to overcome obstacles. Another solution is to use external forces and physical phenomena that occur naturally in the environment around the robot. Movement can also be generated by the transformation of the sphere's shape. Please find below a literature overview as well as a presentation of typical constructions of sphere-like robots.

#### **1.1. Internal cart configuration**

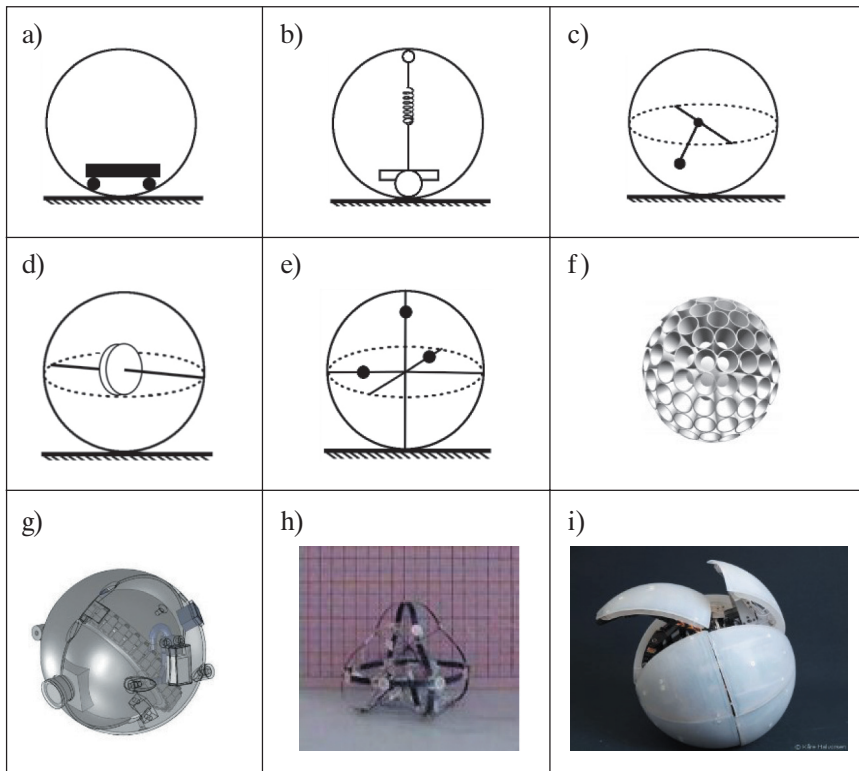
A hamster ball configuration (Fig. 1a) consists of a four-wheel or three-wheel cart placed inside a sphere [7, 4]. Steering can be performed either by changes in velocity or wheel direction. This solution is simple in its design, but it requires uniform sphere; also, the wheels could lose contact with sphere's inner surface due to movement disturbances.

---

\* AGH University of Science and Technology, Department of Automatics and Biomedical Engineering, Krakow, Poland – Ph.D. student

## 1.2. Configuration with internal cart, shaft, and roller

The solution shown in Figure 1b is an extension of the configuration with the cart inside a sphere. A combination of a shaft, spring, and roller allows the cart's wheels to keep contact with the sphere's inner surface. In addition, the shaft balances the construction, and the number of cart wheels can be reduced to two [17, 14]. Omnibola, a noteworthy robot, uses an inverted configuration where the drives are located at the top part of the sphere [11].



**Fig. 1.** Spherical robot design configurations: a) internal cart configuration; b) internal cart with shaft and roller; c) pendulum with an axle; d) flywheel-based design; e) multiple mass-shifting design; f) wind-powered robot [8]; g) underwater robot [12]; h) deformable body design [24]; i) MorpHex [6]

## 1.3. Pendulum with an axle

A configuration with a pendulum (Fig. 1c) requires an axle that connects the sphere to the internal construction. Axle gives a stable basis for the pendulum that can relocate the location of the center of mass and generate both rotation and tilt of the sphere. This

solution restrains the flexibility of the robot, and the sphere's rotation capabilities are limited by the internal axle [4, 9, 15, 25].

#### **1.4. Flywheel-based designs**

This solution uses a momentum or reaction wheel to tilt the internal pendulum or boost the torque of the sphere for a short time (Fig. 1d) [3]. More-advanced designs [4] operate as Control Momentum Gyroscope (CMG) mechanisms, which are usually used to rotate big satellites or space stations.

#### **1.5. Multiple mass-shifting design**

A multiple mass-shifting design uses a collection of actuators for propulsion [13, 4]. The configuration shown in Figure 1e allows for precisely locating the center of mass inside a sphere, but movement generation is complicated and inefficient due to the perpetual operations of multiple actuators.

#### **1.6. Wind-powered robots**

Constructions that take advantage of the environment are strictly limited to specific applications. Control of such a robot requires a mechanism that can redirect external forces and use them to generate movement. The Mars exploration robot research design shown in Figure 1f [8] can be powered by the wind, but the direction of travel cannot oppose wind flow.

#### **1.7. Underwater robot**

An underwater robot is another example of a construction that requires a specific environment in order to work. A spherical robot that uses internal valves and pumps (Fig. 1g) to navigate inside the pipes of nuclear reactor cooling systems was designed to detect radioactive material leaks and corrosion [12].

#### **1.8. Deformable body design**

Shape-memory alloy (SMA) wire is a lightweight material that returns to its original shape when heated. A spherical robot constructed with such a material [24, 4] offers a lot of flexibility; however, movement is unpredictable and difficult to control.

#### **1.9. MorpHex – combination of spherical robot and hexapod**

MorpHex is a robot with multiple legs around its body that can assemble into a ball-shaped construction (Fig. 1i) [6]. In its spherical shape, the robot can still use its legs to push and rotate. Due to the complicated nature of the solution and irregular location

of the robot's legs, motion is constrained in the ball-shaped mode. Moreover, such a construction is vulnerable to hostile environment conditions.

Considering all of the presented solutions listed above, the design based on an internal cart with a shaft and roller offers the best flexibility of movement generation in regards to its complexity. Design of such a construction is the subject of the following paper.

## 2. Design and construction

### 2.1. Mechanical design

Figure 2 shows the basic idea of the robot drive mechanism and design concept. Interior elements of the robot are located inside of sphere (1). The platform (2) is the basis that links all of the elements of mechanical construction. Two DC motors with wheels (6) mounted at the sides of the platform act as a drive mechanism and are in contact with the sphere. The roller (3) has two degrees of freedom and can roll freely along the interior surface of the sphere. The shaft (4) connects the roller and platform, keeping both elements in balance. The spring (5) located at the center of the shaft guarantees that the roller and robot's wheels will never lose contact with the interior surface of the sphere. The platform is a place where the processing unit, control systems, motors, and batteries are located. This concentrates most of the robot's mass, including the center of mass of the construction. The presented configuration allows for modification of the orientation of the interior elements relative to the sphere. At the same time, the location of the center of mass changes, which causes the sphere to roll.

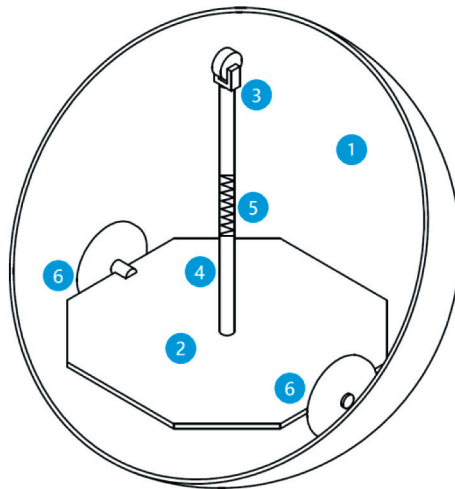
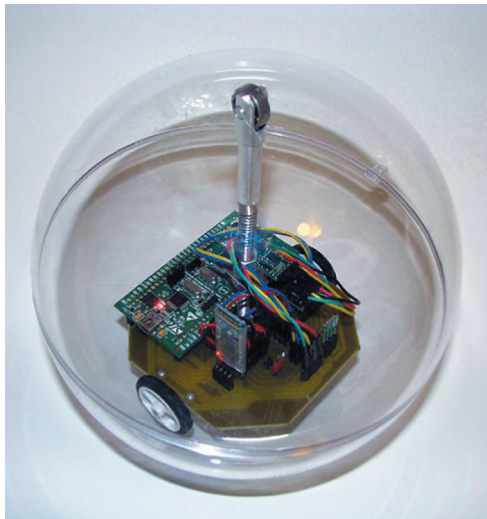


Fig. 2. Robot drive and construction design concept

Figure 3 shows a photo of an assembled spherical mobile robot. A transparent acrylic sphere allows us to observe the behavior of the inner mechanism during movement. The diameter of the sphere is 18 cm, and its wall is 2 mm thick. The drive mechanism is based on two Pololu HP 50:1 DC Motors. They can generate torque up to 0.13 Nm at a maximum rotational velocity of 625/min. The chosen motors can generate torque eight times greater than the torque required to rotate the robot's construction by 180 degrees and overcome friction. Both motors are equipped with 15-mm-diameter wheels and rubber tires. The platform of the robot was designed as a PCB board to provide both a stable basis and necessary circuit connections. The roller has the shape of a closed sleeve that terminates in a yoke, where the axle and bearing acting as a wheel are located. The roller is made of aluminum. Spring parameters were calculated in order to keep the wheels of the robot in contact with the sphere, even in a configuration when robot is upside down. A threaded rod of 6mm diameter was used as a shaft. Thanks to a few additional M6 nuts, it is possible to regulate the length of the shaft and adjust the default position of the spring and, therefore, its stiffness.



**Fig. 3.** Actual photo of assembled spherical mobile robot

## **2.2. Electronic design**

The electronic system of the robot was built with a number of separate modules connected via Inter-Integrated Circuit (I<sup>2</sup>C), Universal Asynchronous Receiver and Transmitter (UART), and General Purpose Input/Output (GPIO) ports. The advantage of this solution is the possibility of easy modifications or module replacement in the event of failure.

Figure 4 shows the relationship between the individual modules. The processing unit of the system is a Cortex-M3-family microcontroller – STM32F100RB [19] operating with a clock speed of 24 MHz. The power supply module consists of two parts; a battery pack and a step-down voltage regulator. This provides power for both the DC motors and the logic units. DC Motor Driver TB6612FNG [22] is the intermediate system between the microcontroller and DC motors. It uses information from the microcontroller and generates PWM control signals enhanced by power directly from the battery pack. Sensor modules consist of a 3-Axis accelerometer LSM303DLHC [21] and 3-Axis gyroscope L3GD20 [20].

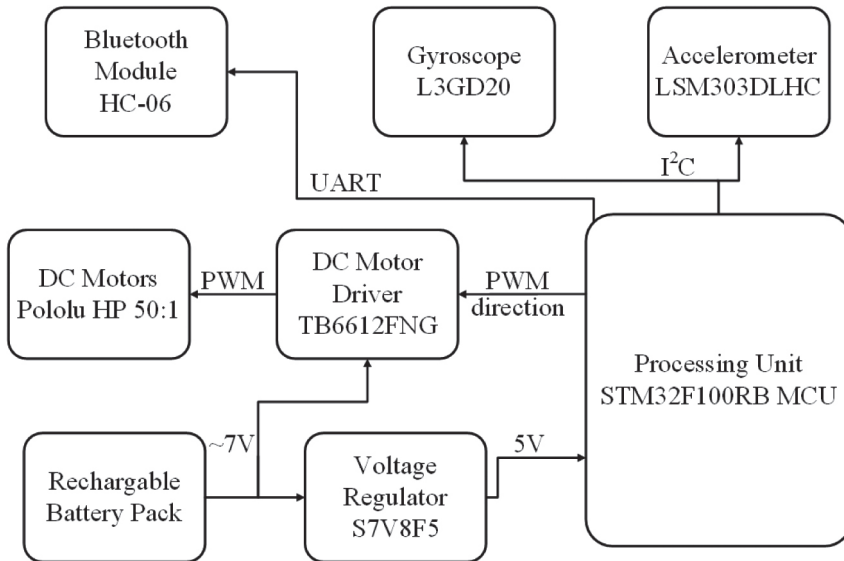


Fig. 4. Hardware module layout and interactions

Information from the sensor modules is collected via an I<sup>2</sup>C bus. The Bluetooth module allows for communication between the robot and external devices. Commands are sent to the robot via a Bluetooth radio signal and transmitted to the microcontroller via a UART bus afterwards. Data transfer also works the other way; messages generated by the microcontroller are delivered to an external device via the communication module.

### 2.3. Sensor overview

To reduce errors associated with linear velocities and accelerations, an accelerometer and gyroscope were placed at a point that remains in close proximity to the center of

the sphere. Figure 5 identifies the parameters measured by the sensors. Axes of the accelerometer and gyroscope are related to the rotation axes of the robot's interior elements. Vectors  $a_x$ ,  $a_y$ , and  $a_z$  represent linear accelerations measured by the accelerometer, and  $g_x$ ,  $g_y$ , and  $g_z$  define actual rotational velocity provided by the gyroscope. An algorithm developed by Sebastian O.H. Madgwick [18, 23] was implemented to estimate the current orientation of the platform. It uses information collected from the sensors and represents orientation as quaternions.

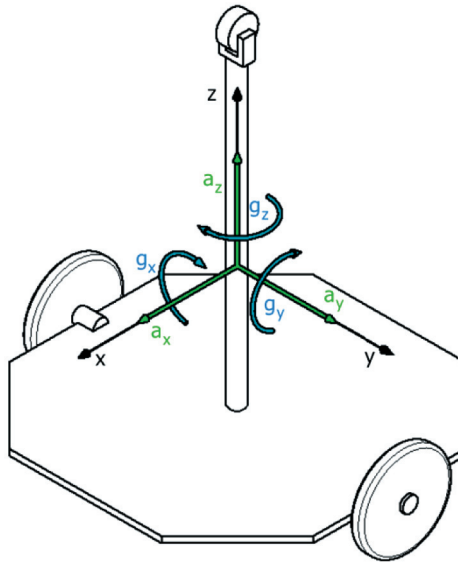


Fig. 5. Sensor location and measured value overview

## 2.4. Communication interface

The data-exchange schema is based on two independent threads of receiving and sending information. The robot does not respond to a received command with a confirmation – it only executes it. Transmission of data to an external device is initialized with the appropriate command that contains information about the requested status variables of the robot. When reporting is enabled, the robot sends data at a constant rate. To disable status reporting, it is necessary to send the appropriate stop command. Such an approach was implemented to increase status report frequency. Configuring the robot to report status of only three state variables at a time allows us to retrieve information with a frequency of 500 Hz. Each command sent to the robot consists of:

- a single byte identifier,
- command parameters,
- 8-bit checksum.

Robot status report frame is of variable length and contains:

- frame length information,
- group of binary flags that indicate which of the status variables are present in the data frame,
- values of state variables present in the data frame,
- 8-bit checksum.

The checksum polynomial is computed in the CRC-8-CCITT standard and is appended to each transmitted message. The probability that false information will not be detected is equal to  $2^{-8}$ .

## **2.5. Software implementation**

Robot controller processes can be split into five main tasks, with higher priority numbers indicating higher priority.

### **2.5.1. Task I, priority 4**

Start-up task – initializes data structures and configures peripherals, sensors, timers, and PWM generator.

### **2.5.2. Task II, priority 3**

Bluetooth communication interrupt handler – writes received data into input buffer and sends data retrieved from output buffer.

### **2.5.3. Task III, priority 1**

Runs a check on whether the input buffer contains a command that can be executed or not, and constructs a status report frame if the output buffer is empty.

### **2.5.4. Task IV, priority 2**

Triggered by the timer each 1 ms, a specific operation order guarantees a constant time interval between sensor reading and an update of PWM signals generator registers, and executes the following operations:

- measures time since last task iteration,
- updates registers of PWM signals generator,
- collects information from sensors,
- estimates orientation of the platform,
- calculates control algorithm,
- stores states and control signals.



### 2.5.5. Task V, hardware generator

Generates PWM signals to the motors. Figure 6 shows a task-sequence diagram. Task IV can be interrupted only by the Bluetooth communication handler (Task II), which is simplified into buffer read and write operations. Such an approach minimizes control sequence delays and improves the quality of the control system.



Fig. 6. Robot controller task sequence diagram

## 3. Mathematical model

### 3.1. Problem overview

Considering a spherical mobile robot as a three-dimensional pendulum suspended at the center of a sphere, construction of a mathematical model is not an easy task and proved to be problematic to solve analytically [4]. However, the trajectory of the sphere can be approximated by a series of short movements in which the sphere changes location in a straight line. In such a situation, a mathematical model can be limited to a two-dimensional scenario, with the pendulum and sphere restricted to movement on a single plane parallel to the force of gravity. A mathematical model for such a simplified problem was derived from Lagrange's equation of motion (1), where  $L$  is the Lagrangian,  $n$  is the number of constraint equations,  $\lambda_i$  are the Lagrange multipliers,  $q_i$  are the generalized coordinates,  $a_{ij}$  are the coefficients in front of  $\dot{q}_i$  of the constraint equations, and  $Q_i$  are the generalized forces or moments [17, 4].

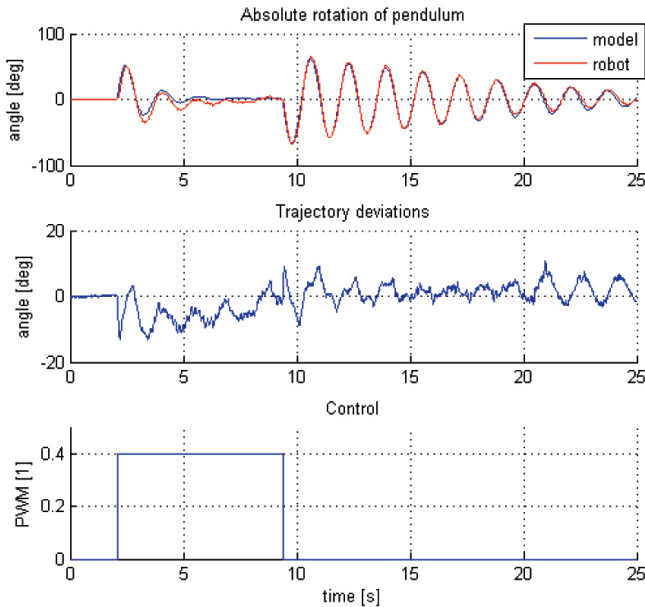
$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum_{j=1}^n \lambda_j a_{ji} + Q_i \quad (1)$$

Identification of the basic model parameters, like the mass of the robot or sphere diameter, was straightforward. However, estimating the location of the center of mass or

friction characteristic required some experiments. Location of the center of mass specifies the length of the pendulum for the mathematical model. Because of the strict relationship between pendulum length and the oscillation period, accurate estimation of the center of mass is crucial. To calculate the location of the center of mass, the robot was placed on an inclined plane in a configuration where the angle of inclination and linear friction allow the robot to remain in one place. In such a configuration, the interior elements of the robot's construction deviate from the regular position. The angle of this deviation was used to calculate the location of the center of mass. Friction modeling along with performed experiments lead to the conclusion that three separate friction models are necessary to properly describe the behavior of the spherical robot: friction model of the rotational movement of the sphere, friction model of the linear movement of the sphere, and friction model of the rotational movement of the pendulum. Friction modeling is a complex issue and will be discussed in a separate paper.

### 3.2. Model-Object comparison

Figure 7 shows the comparison of absolute pendulum rotation trajectories for the mathematical model and spherical robot. The motor control signal takes the form of a short pulse with its PWM duty cycle set to 0.4.



**Fig. 7.** Comparison of absolute pendulum rotation trajectories for mathematical model and spherical robot

## 4. Control system

### 4.1. State variable representation

State variables are defined as shown in Figure 8. State variable representation allows for the easy and intuitive determination of the setpoint value. Variable  $\alpha$  determines the direction of motion as the angle of the range of  $[0, \pi]$ . Variable  $\beta$  is the deviation of the center of mass of the robot from the equilibrium point and takes values in the range of  $[0, \pi/2]$ .

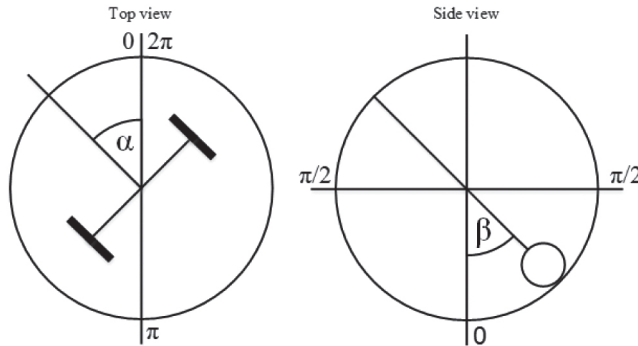


Fig. 8. State variable representation

### 4.2. Calculation of deviations

Calculation of deviations  $\bar{\alpha}$  and  $\bar{\beta}$ , respectively, for  $\alpha$  and  $\beta$  is a complex issue due to their mutual interactions. An important aspect to take into account is the situation in which the modulus of error  $\bar{\alpha}$  is greater than  $\pi/2$ . Then, it is preferable to switch the direction of movement in addition to regular turning. The calculation of deviations  $\bar{\alpha}_i$  and  $\bar{\beta}_i$  with the setpoint values of  $\hat{\alpha}_i$  and  $\hat{\beta}_i$  were carried out by functions (2) and (3).

$$\bar{\alpha}_i = \left( \left( \hat{\alpha}_i - \alpha_i + \frac{\pi}{2} \right) \bmod \pi \right) - \frac{\pi}{2} \tag{2}$$

$$\bar{\beta}_i = \begin{cases} \hat{\beta}_i - \beta_i & 2k\pi \geq \left| \hat{\beta}_i - \beta_i + \frac{\pi}{2} \right| \geq (2k+1)\pi \quad k=0, 1, 2, \dots \\ -\hat{\beta}_i - \beta_i & \text{others} \end{cases} \tag{3}$$

### 4.3. Control algorithm

The control system of the spherical robot contains two PID controllers implemented separately for both  $\alpha$  and  $\beta$  state variables. The controller is defined by the equations (4),

with  $K_p$ ,  $K_I$ , and  $K_D$  as controller parameters,  $U_i$  as control, and  $dt_i$  as step time that denotes the time elapsed between two consecutive executions of Task IV.

$$\begin{aligned}
 U_{\alpha i} &= K_{P\alpha} \bar{\alpha}_i + K_{I\alpha} \sum_{j=0}^i \bar{\alpha}_j dt_j + K_{D\alpha} \frac{\alpha_i - \alpha_{i-1}}{dt_i} \\
 U_{\beta i} &= K_{P\beta} \bar{\beta}_i + K_{I\beta} \sum_{j=0}^i \bar{\beta}_j dt_j + K_{D\beta} \frac{\beta_i - \beta_{i-1}}{dt_i}
 \end{aligned} \tag{4}$$

Control constraints are calculated dynamically in each control system step to ensure that the motor torque is distributed accordingly for both control tasks, change of movement direction, and acceleration. Equation (5) defines control constraints for  $U_{\alpha i}$  and  $U_{\beta i}$ .

$$\begin{aligned}
 U_{\alpha i} \in [-L_\alpha, L_\alpha] \quad L_\alpha &= \frac{|\bar{\alpha}_i|}{|\bar{\alpha}_i| + |\bar{\beta}_i|} \\
 U_{\beta i} \in [-L_\beta, L_\beta] \quad L_\beta &= \frac{|\bar{\beta}_i|}{|\bar{\alpha}_i| + |\bar{\beta}_i|}
 \end{aligned} \tag{5}$$

Calculation of PWM signal generator parameters ( $PWM_{Ai}$ ,  $PWM_{Bi}$ ) is performed with equations (6), parameters were normalized to the values of range  $[-1, 1]$ , negative register setting indicates negative rotation direction of the motor.

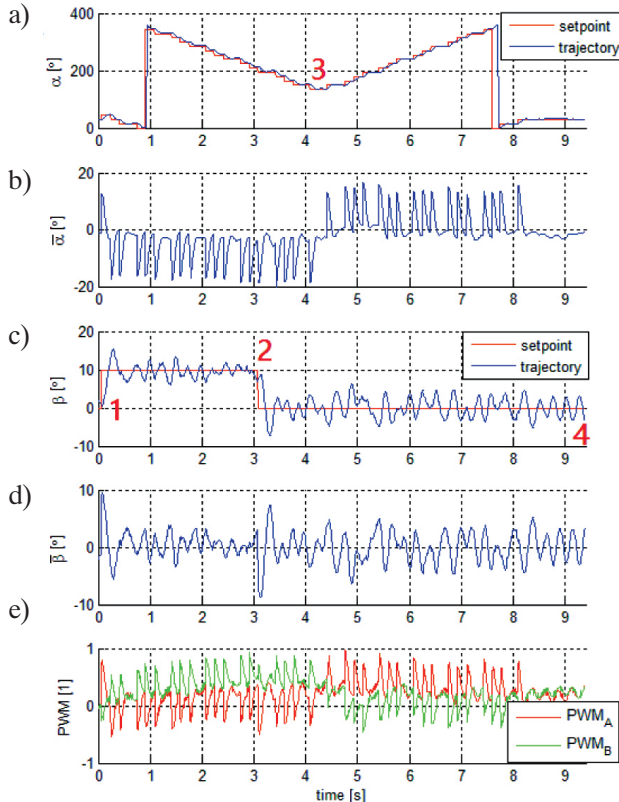
$$\begin{aligned}
 PWM_{Ai} &= U_{\beta i} + U_{\alpha i} & PWM_{Ai} &\in [-1, 1] \\
 PWM_{Bi} &= U_{\beta i} - U_{\alpha i} & PWM_{Bi} &\in [-1, 1]
 \end{aligned} \tag{6}$$

#### 4.4. Trajectory generation

To generate the trajectory of the robot with support from the implemented control system, the trajectories of  $\hat{\alpha}$  and  $\hat{\beta}$  state variables must be specified. The results of the trajectory generation experiment are shown in Figure 9. Data obtained from the sensors do not provide information about the linear translations of the robot. For a more-detailed analysis, the experiment was recorded with a camera. LED markers mounted on the platform allowed us to retrieve information about robot motion trajectory using an image analysis algorithm constructed in MATLAB. Each frame of the recorded movie is transformed into a binary image using a color threshold technique. White-light LED markers are easily distinguished from the background. Figure 10 shows the assembly of each transformed frame of the recording. On the basis of the determined trajectory, it is possible to verify the behavior of the robot in relationship to the supplied control. Key points of

the robot movement trajectory are marked by numbers from 1 to 4. Corresponding changes are also marked in Figure 10. The desired trajectory was generated in parts, and the target point coordinates were transmitted via Bluetooth.

- Point 1 is the starting position of the robot. Initial direction of movement is set to  $\hat{\alpha} = 45^\circ$ . Setpoint value  $\hat{\beta} = 10^\circ$  causes the robot to accelerate over the section of 1–2.
- At point 2, the setpoint value of  $\hat{\beta}$  returns to 0. From this point forward, the task of the  $\beta$ -controller is to maintain the position of equilibrium. Due to the presence of friction, the velocity of the robot begins to decrease.
- In sections 1–3, the trajectory of  $\hat{\alpha}$  starts to change gradually by  $-180^\circ$ , and the robot turns right.
- In sections 3–4, the trajectory of  $\hat{\alpha}$  starts to change gradually by  $180^\circ$ , and the robot turns left.
- At point 4, the robot stops.



**Fig. 9.** Control system trajectories: a) comparison of setpoint and trajectory of  $\alpha$ ; b) deviation of  $\alpha$ ; c) comparison of setpoint and trajectory of  $\beta$ ; d) deviation of  $\beta$ ; e) PWM duty cycle control signals

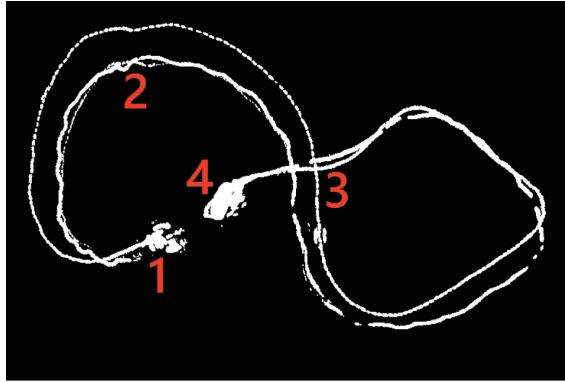


Fig. 10. Recorded motion trajectory of the robot

## 5. Conclusions

The project of designing, constructing, and controlling a spherical mobile robot ended with success. It required a great deal of knowledge from many fields of science, and all of the encountered issues were resolved. The mechanical design was chosen so that the robot could be characterized by multiple degrees of freedom and was able to immediately change its direction of movement. Accelerometer and gyroscope signal analysis allowed us to estimate the orientation of the robot's interior elements. The data retrieved from the sensors proved to be stable enough to use as feedback in the design of a control system for movement-trajectory planning. Hardware and software solutions allowed us to control the robot remotely, design control systems, and gather enormous amounts of data from the experiments. Analysis of the mathematical model provided valuable information in regards to the dynamic of a spherical mobile robot with propulsion based on the relocation of center of mass.

The project of a spherical mobile robot recognized a lot of issues that must be addressed during the design process. The presented construction and test environment can be used for future work on and development of more-advanced control systems. Equipping a robot with additional sensors could also improve the quality of control or redefine the control task.

## References

- [1] Halme A., Schönberg T., Wang Y., *Motion control of a spherical mobile robot*, 4th IEEE International Workshop on Advanced Motion Control, 1996, pp. 100–106.
- [2] Carlsson B., Örbäck P., *Mobile inverted pendulum, report no. ex055/2009*, Technical report, 2009.

- [3] Chemel B., Mutschler E., Schempf H., *Cyclops: miniature robotic reconnaissance system*, IEEE Int. Conf. on Robotics and Automation (ICRA,99), 1999, pp. 2298–2302.
- [4] Schroll G.C., *Dynamic Model of a Spherical Robot From First Principles*, Colorado 2010.
- [5] Alemi Ardakani H., Bridges T.J., *Review of the 3-2-1 Euler Angles*, University of Surrey, Guildford 2010.
- [6] Halvorsen K., *Morphex*, <http://www.robotee.com/index.php/robotprojects/morphex-project-morphing-hexapod-by-kare-halvorsen-part-2/>, 2015.
- [7] Alves J., Dias J., *Design and control of a spherical mobile robot*, Technical report, Coimbra 2003.
- [8] Antol J., Kolacinski R., Minton D., Owens R., Parker J., *Low cost mars surface exploration: the mars tumbleweed*, Technical report, NASA Langley Research Center, 2003.
- [9] Laplante J., Masson P., Michaud F., *Analytical longitudinal and lateral models of a spherical rolling robot*, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5715&rep=rep1&type=pdf>, 2007.
- [10] McLean K., *Mclean monocycle*, <http://www.mcleanmonocycle.com/>, 2014.
- [11] Jaimez M., Castillo J.J., Garcia F., Cabrera J.A., *Design and modelling of Omnibola, a spherical mobile robot*, Mechanics Based Design of Structures and Machines 2012, Vol. 40(4), pp. 383–399.
- [12] MIT, *A compact underwater vehicle using high-bandwidth coanda-effect valves for low speed precision maneuvering in cluttered environments*, <http://newsoffice.mit.edu/2011/nuclear-robots-0721>, 2014.
- [13] Mukherjee R., *Design Challenges in the Development of a Spherical Mobile Robot*, Robotic Ball Technology Study for Planetary Surface Missions. NASA JSC/EV George Studor, 2010.
- [14] Orbotix Inc. Sphero, <http://www.gosphero.com/>, 2014.
- [15] Rotundus. Groundbot, <http://www.rotundus.se/>, 2014.
- [16] Bhattacharya S., Agrawai S., *Experiments and motion planning of a spherical rolling robot*, Technical report, April 2000.
- [17] Firlej S., Rozycki M., *Design and construction of a spherical mobile robot*, AGH University of Science and Technology, Poland, Master of Science Thesis, Supervisor: A.K. Pilat, Krakow 2013.
- [18] Madgwick S., *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*, Department of Mechanical Engineering, 2010.
- [19] STMicroelectronics. *STM32F100 Advanced ARM-based 32-bit MCU (Doc ID 16455 Rev 5)*, 2010.
- [20] STMicroelectronics. *L3GD20 three-axis digital output gyroscope (Doc ID 022116 Rev 1)*, 2011.
- [21] STMicroelectronics, *LSM303DLHC Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module (Doc ID 018771 Rev 1)*, 2011.
- [22] Toshiba, *TB6612FNG Driver IC for dual DC motor*, 2008.
- [23] X-IO Technologies. *Open source AHRS*, <https://github.com/xiotechnologies/open-source-ahrs-with-x-imu>, 2013.
- [24] Sugiyama Y., Shiotsu A., Yamanaka M., Hirai S., *Circular/spherical robots for crawling and jumping*, Technical report, April 2005.
- [25] Landa K., Pilat A.K., *Design and start-up of spherical robot with internal pendulum*, RoMoCo'15, 10<sup>th</sup> International workshop on Robot Motion and Control: 6–8 July 2015, Poznań, Poland, pp. 27–32, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp==7219709>.