

Article

Shape Sensing of Hyper-Redundant Robots Using an AHRS IMU Sensor Network

Ciprian Lapusan , Olimpiu Hancu *  and Ciprian Rad 

Department of Mechatronics and Machine Dynamics, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; ciprian.lapusan@mdm.utcluj.ro (C.L.); ciprian.rad@mdm.utcluj.ro (C.R.)

* Correspondence: olimpiu.hancu@mdm.utcluj.ro

Abstract: The paper proposes a novel approach for shape sensing of hyper-redundant robots based on an AHRS IMU sensor network embedded into the structure of the robot. The proposed approach uses the data from the sensor network to directly calculate the kinematic parameters of the robot in modules operational space reducing thus the computational time and facilitating implementation of advanced real-time feedback system for shape sensing. In the paper the method is applied for shape sensing and pose estimation of an articulated joint-based hyper-redundant robot with identical 2-DoF modules serially connected. Using a testing method based on HIL techniques the authors validate the computed kinematic model and the computed shape of the robot prototype. A second testing method is used to validate the end effector pose using an external sensory system. The experimental results obtained demonstrate the feasibility of using this type of sensor network and the effectiveness of the proposed shape sensing approach for hyper-redundant robots.

Keywords: sensor network; IMU sensor; robot shape sensing



Citation: Lapusan, C.; Hancu, O.; Rad, C. Shape Sensing of Hyper-Redundant Robots Using an AHRS IMU Sensor Network. *Sensors* **2022**, *22*, 373. <https://doi.org/10.3390/s22010373>

Academic Editor: Ciprian Dobre

Received: 18 November 2021

Accepted: 2 January 2022

Published: 4 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hyper redundant robots are characterized by the ability to execute complex movements in workspaces with obstacles due to the large number of Degrees of Freedom (DoF). Usually, the number of DoF that characterized such a robot is $\gg 6$, typically above 12-DoF. As a result, the robot has the possibility to reach a certain position in the workspace in an almost infinite number of configurations for the joints values [1].

These advantages make this type of robots very appealing in certain fields where a high degree of flexibility and adaptability is required during the operations. The development of hyper-redundant robots can be traced back to 1972 when Hirose et al. built a snake-like robot prototype with 20-DoF called ACM III [2]. That prototype became the reference for this type of robots and many researchers tried over the years to improve the original concept in different ways [3].

Structurally, snake-like hyper-redundant robots (serpentine robots) can be categorized as (1) continuum/soft manipulators with an “infinite number of DoF” with a single continuous flexible backbone and (2) articulated joint-based rigid manipulators with a limited number of DoF composed of modules serially connected in hyper-redundant structures that can produce smooth curves [4,5].

In the last years, these robots were increasingly used as advanced dexterous manipulators for various tasks in different domains. For example, OC Robotics has built Series II—X125 snake-arm industrial robot that has 12 links serially connected each with 2-DoF for a total of 24-DoF. It has a total bending capability of 225 deg with a 27.5 deg per link and can manipulate payloads up to 10 kg. The robot can be used for cutting, welding, grasping, or fastening operations [6]. More recently, Martín-Barrio et al. [7] designed a discrete joint-based hyper-redundant cable-actuated robot with 7 modules serially connected for a total of 14-DoF to be used for inspection tasks in constrained environments. Shape sensing, kinematic control and remote operation in an immersive reality were the main challenges

reported by the authors. Precision/vertical farming is another field with unstructured environment where hyper-redundant robots are increasingly used for soft grasping, harvesting, precision planting, precision spraying, or precision fertilizing tasks [8–10]. However, design, control and shape sensing are the main challenges when it comes to implement these types of manipulators in this area [11].

Hyper-redundant robots have great potential to be used in unstructured environments where highly dexterous manipulations are needed and although these robots have been used in many applications in the last decade, precise modelling, kinematic control, and mechanical design are still challenging problems as highlighted in [12–15].

A practical solution in this matter is to use an advanced real-time feedback system for shape sensing. That system should include advanced shape sensing sensors, model-based shape reconstruction techniques (kinematic models such as curve parametrization and Piecewise Constant Curvature and/or dynamic models such as Cosserat Rod, spring-mass model, and real-time finite element method [1]), robust communication protocols and data fusion algorithms to name a few. In this context, this paper addresses the issues of shape sensing for joint-based snake-like hyper-redundant robots.

As highlighted in [16,17], for shape sensing there are two types of sensors: non-contact-based (NCSSs) and contact-based shape sensors (CSSs). NCSSs include video cameras with imaging techniques such as Fluoroscopy, RaDAR and LiDAR technologies but it is well known that ambient temperature and environment characteristics are directly influencing the response of these sensors.

On the other side, CSSs include encoders, electrical resistivity and strain sensors, capacitive flexible sensors, optoelectronics sensors, fiber optic sensors (FOSSs) and Micro/Nano Electro-Mechanical Systems (MEMS/NEMS) Inertial Measurement Unit (IMU) sensors. Encoders are well established solutions that provide the angular movement of joints but if compact designs are needed, their integration could be a major problem. Moreover, if the hyper-redundant robot has many DoF and encoders need to have a high number of pulses per turn, the cost of such a solution can increase drastically [18]. Electrical resistivity and strain sensors are more suited for gaming and wearables technology, require many cables, and their installation can be costly [19]. An alternative technology is that of flexible soft sensors. For example, Bend Labs [20] developed a set of flexible sensors based on differential capacitance measurement of angular displacement. They claim their pioneering technology is capable of sub-degree resolution and does not suffer from drift. Optoelectronic sensors can sense the shape in real-time and are a combination of light sensors, gyroscopes, and accelerometers [21]. For example, Dementyev et al. [22] developed a low-cost contact-based optoelectronic shape sensor called SensorTape suited for posture measurement. FOSSs is an emerging sensing technology that enables 3D shape reconstruction by using multi-core optical fiber cable together with strain sensors with a submillimeter accuracy [16,17]. As an example, Schmitz et al. presented in [23] an articulated single joint surgery snake-like robot prototype with applications in minimally invasive surgery. Their results indicate a mean absolute error (MAE) of 0.71 deg over a 35 deg bending range. Although FOSSs is a very promising shape sensing technology, the majority of works present proof-of-concept solutions in medical/geotechnical applications, and it is well accepted in the literature that many technical issues must be overcome to become a mature technology. For a comprehensive review of FOSSs and its applications the reader should consult [16,17].

MEMS/NEMS IMU sensors have become very accessible in the last years due to the miniaturization and advances in MEMS/NEMS technologies and are increasingly used as shape sensing solutions in many robotic applications [24–26]. For example, one encoder can be replaced by an IMU sensor reducing the cost up to five times [18]. IMU sensors can be found in packages with 6-DoF (a 3-axis accelerometer and a 3-axis gyroscope) or 9-DoF (combined with an extra 3-axis magnetometer to work as an Attitude and Heading Reference System (AHRS) [27]). However, integration drift, magnetic disturbances, electro-magnetic noise, and calibration procedures are the main issues when it comes to IMUs, but such problems are usually solved by (1) fusing inertial measurements with

additional information from kinematic mathematical models and (2) by using extensive signal processing and well-established digital filtering techniques such as Extended Kalman Filtering (EKF) or Unscented Kalman Filtering (UKF) [28–32]. Thus, stable data output can be measured with a precision as high as 0.05 deg for X-Y axes and 1 deg for Z axis as reported in [33]. In addition, their reliability was confirmed in studies such as barbell velocity assessments [34], pervasive healthcare applications [35], online kinematic calibration of robot manipulators [36] or wireless body area networks [37].

Concerning the use of IMUs in snake-like hyper-redundant robots, Kang et al. [38] validated a closed-form kinematic model of a continuum robot actuated pneumatically using an Xsens Mti-30 IMU sensor attached to the end-effector. The results showed that pitch and roll angle were estimated by the model with an error of 1 deg. In [39], Peng et al. developed a three-section continuum manipulator (lower, middle, and upper joint) actuated by nine pneumatic artificial muscles. To validate the kinematic and compliance models they used three 6-DOF IMUs attached at the end of each section to measure bending angle, acceleration, and rotation parameters. The errors obtained at the end-effector were less than 7 mm, one source of these errors was related with sensors measurements as highlighted by the authors. In [40], Zhang et al. developed a snake-like robot prototype with micro-inertial sensors. They attached a BSN-Body Sensor Network (a 3D accelerometer ADXL330 and a 3D gyroscope ITG-3200) on each segment of the robot and used a proprietary algorithm to estimate joint angles accurately. The readings from BSNs were compared with the reading from the on-board encoders. A Root Mean Square Error (RMSE) error between 0.4–0.47 deg and between 1.17–1.22 deg was reported for yaw and pitch angle, respectively, for different elements of the robot. In [41], Luo et al. proposed a fusion pose estimation method (RBF-UKF) for a redundant robot that is based on a multisensory fusion approach that is applied in two phases: a “pre-enhancement” fusion phase where information from a RGB-D camera and a MARG (Magnetic, Angular Rate, and Gravity) sensor are fused with the information from an optical encoder and an adaptive fusion phase where the pose of the robot is predicted and various parameters are adaptively adjusted. Their experimental setup consists of eight modules with 1-DoF serially connected as a redundant manipulator. Experimental results showed that RMSE error for pose estimation with EKF and UKF filtering methods were four times higher than their proposed RBK-UKF method. In another paper, Zhao et al. [42] proposed an autonomous navigation method for a joint-based snake-like robot considering robot’s motion characteristics, a strapdown inertial navigation system, and sensor fusion techniques using EKF filtering. Various experiments were conducted (from linear motion to circular motion) and position error was less than 5% of total traveled distance of the robot.

As can be seen from the analysis, various sensing technologies can be used for hyper-redundant robots, but IMUs is highlighted as a promising solution in many research papers. Therefore, although IMUs lead to larger errors than those obtained with encoders, advantages such as miniaturization, low energy consumption, reduced costs, and small weight, make IMUs an appropriate candidate for certain applications of snake-like hyper-redundant robots where the problem of shape control is important for navigation in unstructured environments. If the end effector of the robot integrates an additional motion sensor that helps to maintain the robot’s end effector on the planned trajectory, a deviation from the planned shape can be accepted (with a certain margin of error).

In this context, the paper addresses the problem of shape sensing and pose estimation of an articulated joint-based snake-type robot (called Python), that is designed to operate in unstructured environments under various uncertainties. The original contributions are related to (1) the robots shape sensing computational system and (2) the Hard-ware-in-the-Loop (HIL) testing method.

The shape sensing computational system is a proprietary algorithm custom designed for a Python robotic system but can be used also for other hyper-redundant robots. It addresses the problem of kinematics and shape sensing of the robot (using a network of IMU AHRS), as a part of the control system useful in path planning and navigation

tasks (in unstructured environments). The computational system integrates a dedicated network of AHRS IMU sensors and a shape sensing algorithm that runs in real time on a microcontroller-based board. The IMU sensor network is embedded into the structure of the robot and communicates using CAN network, thus resulting in a compact design with a reduced number of communication wires. The proposed sensing algorithm uses the data from the sensor network to directly calculate the kinematic parameters of the robot in the modules' operational space reducing in this way the computational time.

The novel testing method is based on the HIL technology and allows real-time testing and assessment of algorithms used for calculus of robot shape and pose for hyper-redundant robots that integrate a sensory network. The method uses the information from the sensors of a real robot, but the estimation of shape and position are made using a virtual model/robot. The same information is entered into the robot's position and shape computational algorithm, and the results are compared. Once validated, the algorithm could be used by the real robot as part of the robot's control strategy.

The advantage of this method is that the shape algorithm validation could be performed without the need of an external sensory system to measure the pose of each robot platform/module. The flexibility of the Simscape platform also allows for the implementation of different robotic structures, thus the method could be applied to other similar robot topologies. Extending this setup, the method could also be used to test the entire control strategy of the robot, benefiting from the advantages of HIL simulations.

The paper is structured as follows. Chapter 2 presents the conceptual design of the Python robot that can be used for dexterous manipulations in unstructured environments, while in Chapter 3 details are given about the advanced real-time feedback system integrated in Python for shape sensing and pose estimation. A HIL Simulation scenario is presented in Chapter 4 where the experimental results from the experimental setup are compared with the simulation result of a virtual model of the robot that is running in parallel on a dSPACE HIL simulator. Finally, the paper ends with the conclusion.

2. Robot Structure Description and Kinematics

The Python robot has a hyper-redundant structure that consists of a set of n ($n = 10 \div 20$) identical modules that are serially connected (Figure 1a). On the last module a soft gripper and/or an inspection camera can be connected, allowing thus the implementation of tasks related to harvesting and inspection for a predefined range of crops cultivated in vertical farms.

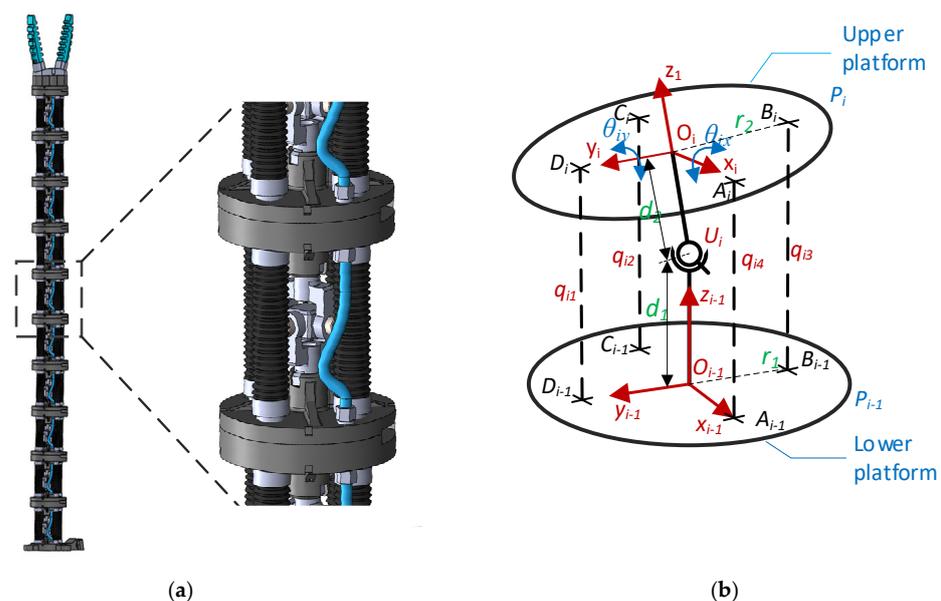


Figure 1. Python Robot (a) CAD Model with $n = 10$ modules (b) kinematic diagram of one module.

Each module of the robot has two platforms that are connected through a universal joint U_i $\{i = 1 \dots n\}$ that is at the distance d_1 from the lower platform and d_2 from the upper platform. The actuation of the structure is performed by four bellows actuators that are working in tandem (two by two). The actuators are connected on the lower and upper platforms on the positions indicated by the points A_i/A_{i-1} , B_i/B_{i-1} , C_i/C_{i-1} and D_i/D_{i-1} (Figure 1b). Thus, each module has 2-DoF (two rotations along the axis O_{ix} and O_{iy} $\{i = 1 \dots n\}$), respectively, for $n = 10$ the whole structure will have 20-DoF. The end effector position and orientation are characterized by the following parameters: G_x , G_y , G_z and G_α , G_β , G_γ .

In order to control the robot behavior and implement the path planning strategies the kinematic parameters of the entire structure must be determined (robot shape sensing). For this type of robots, the kinematics can be expressed as a successive mapping between actuator space, module operational space and robot operational space [43,44] (Figure 2). In the paper a new approach is proposed, as presented in [45] by the authors, where the kinematic is implemented by mapping only between module operational space and robot operational space. This is achieved by fusing the data from the robot sensor network and determining the kinematic parameters of the upper platforms in each module directly for all the robot's modules.

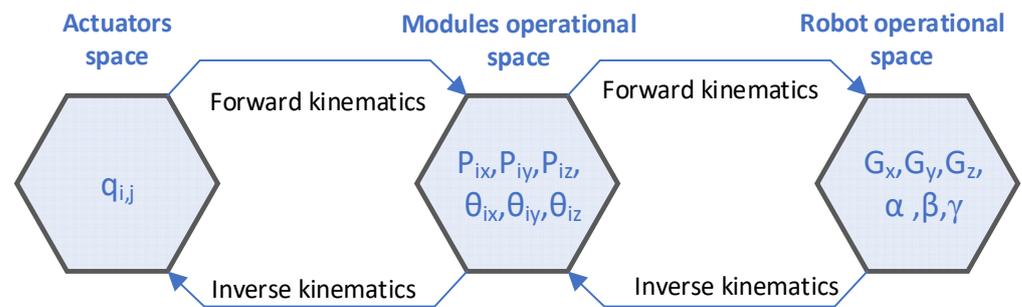


Figure 2. Kinematic analysis of hyper-redundant robots [43].

The proposed method uses the orientation data for the upper and lower platform, that are provided by the sensory system and the geometric characteristics of the modules to calculate the direct kinematics of the robot. The algorithm receives the Euler angles α_i , β_i and γ_i $\{i = 0 \dots n\}$ that define the orientation of all the platforms P_i $\{i = 0 \dots n\}$. The rotation sequence that defines the orientation of each platform is z - y - x . The rotation matrix for each platform is determined by using (1).

$$R_{P_i} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (i = 0 \dots n) \quad (1)$$

For each R_{P_i} $\{i = 0 \dots n\}$ matrix, the r_{kl} $\{k = 1 \dots 3; l = 1 \dots 3\}$ components are calculated based on the α_i , β_i and γ_i $\{i = 0 \dots n\}$ Euler angles, using the following equations:

$$\begin{aligned} r_{11} &= \cos(\gamma_i) \cos(\beta_i) \\ r_{12} &= \cos(\gamma_i) \sin(\beta_i) \sin(\alpha_i) - \sin(\gamma_i) \cos(\alpha_i) \\ r_{13} &= \cos(\gamma_i) \sin(\beta_i) \cos(\alpha_i) + \sin(\gamma_i) \sin(\alpha_i) \\ r_{21} &= \sin(\gamma_i) \cos(\beta_i) \\ r_{22} &= \sin(\gamma_i) \sin(\beta_i) \sin(\alpha_i) + \cos(\gamma_i) \cos(\alpha_i) \\ r_{23} &= \sin(\gamma_i) \sin(\beta_i) \cos(\alpha_i) - \cos(\gamma_i) \sin(\alpha_i) \\ r_{31} &= -\sin(\beta_i) \\ r_{32} &= \cos(\beta_i) \sin(\alpha_i) \\ r_{33} &= \cos(\beta_i) \cos(\alpha_i) \end{aligned} \quad (2)$$

The R_{P_i} $\{i = 0 \dots n\}$ rotation matrices are used to calculate R_{R_j} $\{i = 1 \dots n\}$ rotation matrices that defines the relative rotation between two consecutive platforms (P_j and P_{j-1}).

$$R_{R_j} = R_{P_j} * transpose(R_{P_{j-1}}) ; \{j = 1 \dots n\} \quad (3)$$

The R_{R_j} $\{i = 1 \dots n\}$ rotation matrices are then used to extract the relative rotation angles θ_{jx} and θ_{jy} between two consecutive platforms P_j and P_{j-1} $\{j = 1 \dots n\}$ (Figure 1b). Due to the universal joint between the two platforms, the relative rotation θ_{jz} along the axis O_{jz} is always zero. The θ_{jx} and θ_{jy} angles are calculated using the following equations:

$$\begin{aligned} \theta_{jy} &= atan2\left(-R_{R_j}(3,1), \sqrt{R_{R_j}(1,1)^2 + R_{R_j}(2,1)^2}\right) \\ \theta_{jx} &= atan2\left(\frac{R_{R_j}(3,2)}{\cos(\theta_{ji})}, \frac{R_{R_j}(3,3)}{\cos(\theta_{ji})}\right) \end{aligned} \quad (4)$$

where: $R_{R_j}(1,1)$, $R_{R_j}(2,1)$, $R_{R_j}(3,1)$, $R_{R_j}(3,2)$ and $R_{R_j}(3,3)$ are the elements of R_{R_j} rotation matrix.

Using the robot geometric dimensions and the relative angles between two consecutive robot platforms, the position and orientation of all mobile platforms P_j $\{j = 1 \dots n\}$ can be calculated with respect to the world frame. For this purpose, an iterative process, starting from the base (platform P_0) is implemented using a set of homogenous transformations. The following general equation is used:

$${}^j_0T = {}^{j-1}_0T {}^j_{j-1}T \quad \{j = 1 \dots n\} \quad (5)$$

At each module level (two consecutive platforms), the homogenous transformation matrix ${}^j_{j-1}T$ includes one translation (along O_{jz} axis with d_1), two orientations (along O_{jx} and O_{jy} axis with θ_{jx} and θ_{jy} angles) and one translation (along O_{jz} axis with d_2). Equation (6) describes this process:

$${}^j_{j-1}T = \begin{bmatrix} 1 & 0 & 0 & P_{d1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{\theta_{jx}, \theta_{jy}} & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 & P_{d2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Based on the homogenous transformations j_0T for P_j $\{j = 1 \dots n\}$ platform, the position and orientation relative to the global coordinate system can be expressed using (7) and (8).

$$\begin{aligned} P_{jx} &= {}^j_0T(1,4) \\ P_{jy} &= {}^j_0T(2,4) \\ P_{jz} &= {}^j_0T(3,4) \end{aligned} \quad (7)$$

$$\begin{aligned} P_{j\alpha} &= atan2\left(\frac{{}^j_0T(3,2)}{\cos(P_{j\beta})}, \frac{{}^j_0T(3,3)}{\cos(P_{j\beta})}\right) \\ P_{j\beta} &= atan2\left(-{}^j_0T(3,1), \sqrt{\left({}^j_0T(1,1)\right)^2 + \left({}^j_0T(2,1)\right)^2}\right) \\ P_{j\gamma} &= atan2\left(\frac{{}^j_0T(2,1)}{\cos(P_{j\beta})}, \frac{{}^j_0T(1,1)}{\cos(P_{j\beta})}\right) \end{aligned} \quad (8)$$

The end effector pose is calculated using the n_0T_G (9) homogenous transformation matrix in the same way as presented above. The pose of the end effector characteristic point G (G_x , G_y , G_z and G_α , G_β , G_γ) relative to P_n platform, is determined by adding

additional translation and/or rotations T_G depending on the geometric characteristics of the gripper/camera that is connected to the robot.

$${}^n T_G = {}^n T T_G \quad (9)$$

The presented algorithm is next implemented in MATLAB/Simulink and the generated code runs on Discovery microcontroller-based board as part of the proposed shape sensing computational system. The algorithm runs in real time and the results are displayed on a dedicated Graphical User Interface (GUI).

3. Design and Implementation of the Shape Sensing Computational System

The main components of the shape sensing computational system are presented in Figure 3. The system integrates: the dedicated network of AHRS IMU sensors; the shape sensing algorithm that runs in real time on the Discovery board and the robot GUI.

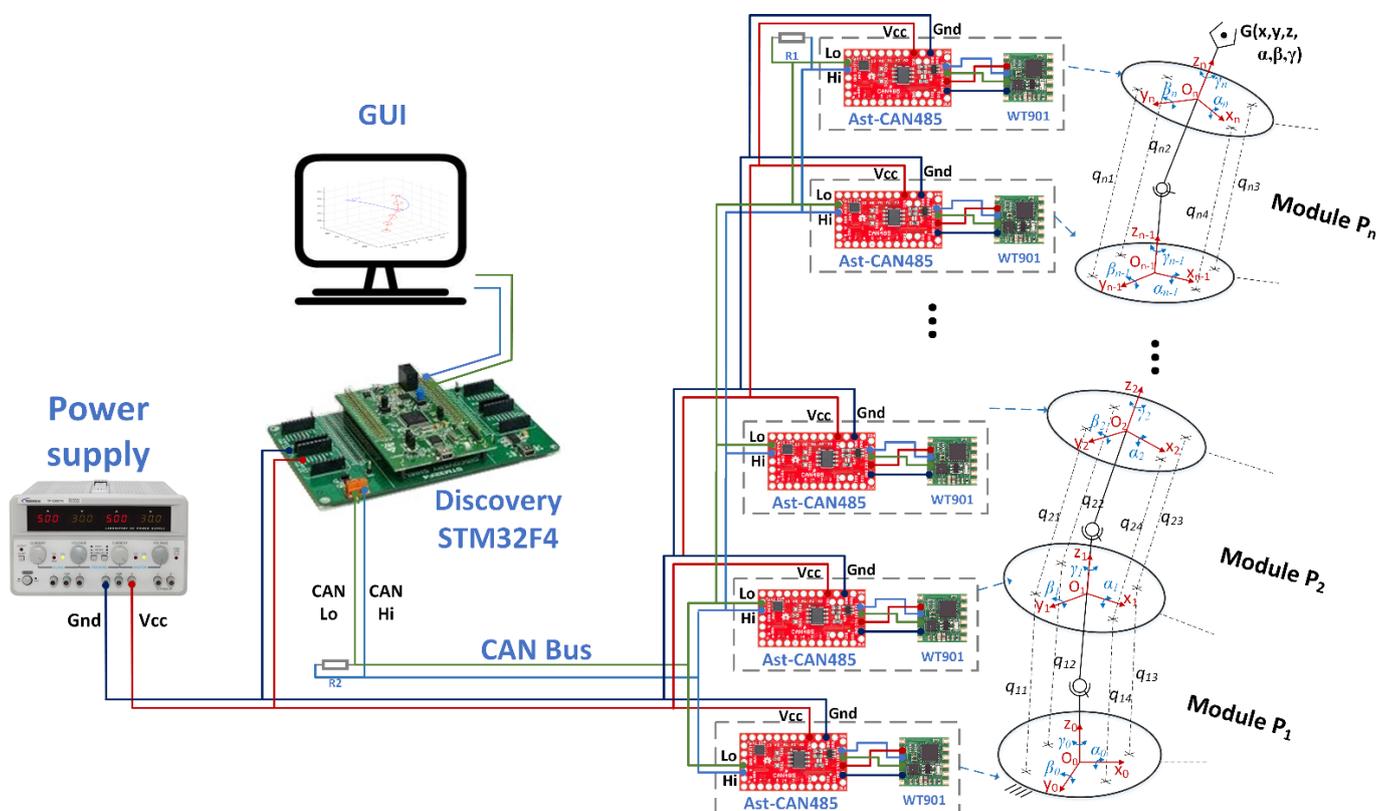


Figure 3. Schematic of the shape sensing computational system.

The sensor network includes the nine-axis attitude angle sensors (model WT901 produced by WitMotion [33]) that are attached to each robot module. Each AHRS IMU sensor is connected to an AstCAN 485 microcontroller board and the communication between the two components is implemented using the UART interface. All the AstCAN boards are interconnected using a CAN network, allowing exchange in real time of information between the central control system (Discovery board) and each module. The sensors provide the orientation of each platform (the attitude angles: α_i , β_i and γ_i $\{i = 0 \dots n\}$) using the Northeast sky coordinate system [33]. The WT901 sensors are used in the proposed shape sensing computational system due to their high precision in measuring the angles along X and Y axis of 0.05 deg. The precision along Z axis is 1 deg and the maximum data output frequency from each sensor is 100 Hz. For calculating the attitude, the IMU sensor integrates a high-precision gyroscope, an accelerometer and a geomagnetic sensor. The accelerometer range was set to 16 g and the gyroscope maximum speed was set to

2000 deg/s. The sensors data are locally fused using a dynamic Kalman filter algorithm with a bandwidth of 20 Hz.

The sensor data, provided by each robot module, are further processed by the Discovery board that runs the proposed shape-sensing algorithm. The algorithm uses the sensors data to detect the shape and pose of the end effector. The software was developed using MATLAB/Simulink, and the code was generated automatically using Simulink Coder. The execution time of the code is less than 50 ms and the main program sampling period is 20 Hz. The obtained numerical results are displayed on the robot GUI that runs on a PC. The GUI was developed using MATLAB Guide tool and communicates with the Discovery board using the UART interface.

For testing the proposed shape sensing computational system, an experimental stand was developed as can be seen in Figure 4. The stand includes: the prototype of the proposed hyper-redundant robot with a total of 8-DoF (number of modules $n = 4$), the shape sensing computation system previously described and the Patriot Polhemus Sensory System (used for testing purposes). As mentioned before, for each module a WT901 attitude sensor is attached to the lower and upper platform resulting in a total of 5 AHRS IMU sensors embedded in the structure of robot prototype, which are used to determine the shape of the robot.

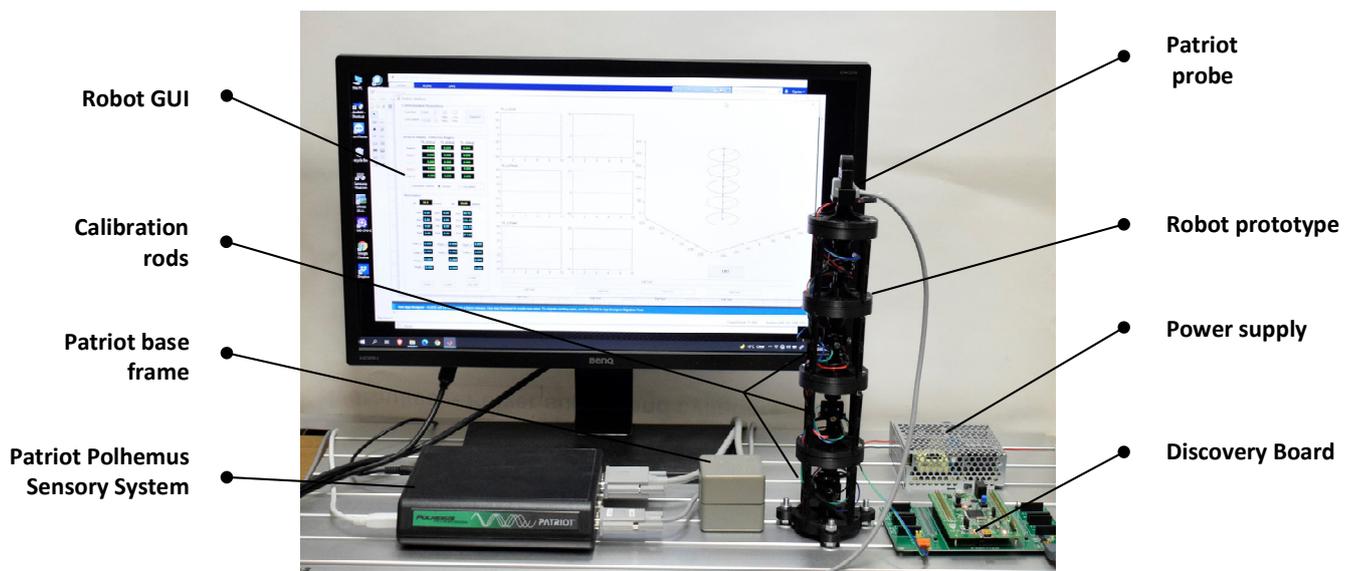


Figure 4. Experimental Stand—Robot prototype with $n = 4$ modules.

The calibration process of the system takes into account the calibration of each IMU sensor used in the sensory network and the calibration of the robot structure after mounting all the sensors. The IMU calibration is performed before mounting on the robot and consists in accelerometer calibration and magnetic calibration based on the methodology described in the WT901 sensor manual [33]. The robot calibration aims at eliminating any mounting errors of the sensors. Therefore, a set of rods were designed and mounted on the robot structure during the calibration process. The rods position each robot platform parallel to the base frame (as presented in Figure 4). In this position the roll and pitch should be zero and the differences recorded from the sensor data are stored and added as bias by the Discovery board. The alignment along z axis is also analyzed and calibrated.

4. Experimental Results

4.1. Testing Methods

The effectiveness and performance of the robot shape sensing computational system is evaluated using two testing methods. The first method is based on HIL technology and

uses a virtual model to validate the shape sensing algorithm (robot shape, end-effector pose). The second method uses an external sensory system to validate the output of the shape sensing computational system in terms of end-effector pose.

The first testing method uses a HIL technology to evaluate the shape sensing algorithm. HIL (Hardware-in-the-loop) simulations are defined as synergetic combinations between physical and virtual components which allow development of experiments where real and virtual components run in the same application [46]. The evaluation of the efficiency (precision) of the shape sensing algorithm is important because it is the basis of the control strategy (as part of the control system). In the proposed method the data provided by the robot sensors are used to detect the shape of the robot by the shape sensing algorithm which runs on a Discovery board. The same data are transmitted to a virtual model (that runs on dSPACE) which is used to validate the effectiveness of the algorithm. If large errors occur between these two approaches, they can only be caused by the shape sensing algorithm because the virtual model is built based on a certified modeling technology (MATLAB/Simulink) and simulated on a real-time platform (dSPACE). Small errors may occur due to different discretization or different calculation precision specific to simulation platforms (Discovery vs. dSPACE).

A conceptual diagram for the proposed testing method setup is presented in Figure 5. In this experimental setup the robot prototype is running in parallel with a virtual robot model that is implemented on dSPACE DS1006 HIL Platform and the numerical results obtained are analyzed and compared. In order to evaluate the precision of the shape sensing computational system, the MAE error is calculated.

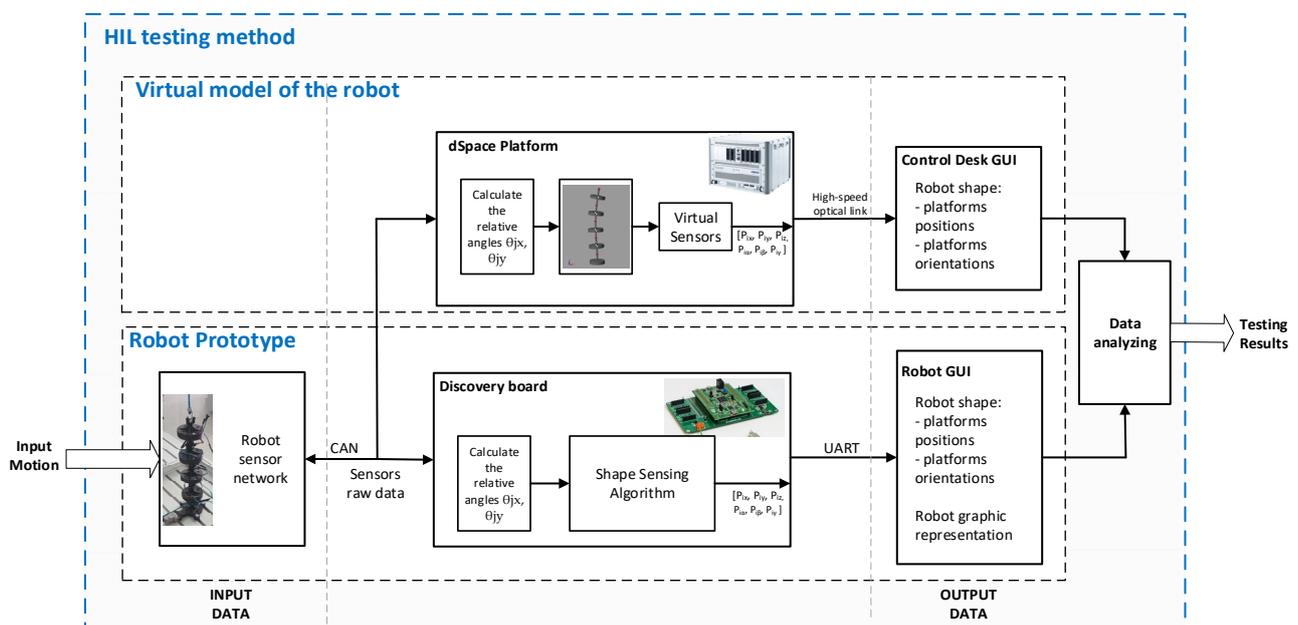


Figure 5. Conceptual diagram for the proposed testing method.

The data processing flow in the proposed method is as follows. The raw data (Input Data) provided by sensors are used as inputs in the virtual model which return the position and orientation of each platform through virtual sensors and implicit kinematics of the virtual robot model. The virtual model of the robot is implemented using Simulink/Simulink blocks and replicates the prototype of the real robot. As part of the model, dedicated virtual sensor blocks for each platform that form the modules of the robot are implemented. Thus, the correct readings on the platform's position and orientation are obtained. The results (Output Data) are displayed using ControlDesk (dSPACE GUI interface). The Output Data calculated by using the proposed shape sensing algorithm that runs on Discovery board are analyzed and compared with the ones obtained from the virtual robot that runs on the

dSPACE platform. The obtained data are saved using the GUI interfaces and the results are analyzed in MATLAB.

The second testing method is used to validate the shape sensing computational system by measuring the end-effector pose. The conceptual diagram of the method is presented in Figure 6. In this setup the Patriot Polhemus sensory system is used to measure the end effector position and orientation. For measuring these parameters, the Patriot sensory system uses one base frame (electromagnetic field source) that is attached to the fixed frame of the robot and a probe that is mounted on the end effector on the robot. The sensors' base frame produces its own electromagnetic field that is used to calculate the probe's relative distance and orientation. The sensor provides data with a resolution of 0.08 mm and 0.016 deg at a 60 Hz rate for measurement setups where the distance between the probe and fixed base is less than 609 mm (in this setup the average distance was 350 mm between the two elements) [47]. The data provided by the Patriot sensory system (parameters: sG_x , sG_y , G_z , sG_α , sG_β , sG_γ) are displayed and stored using the Patriot GUI interface. These data are then compared with the values calculated by using the proposed sensing computational system (parameters: G_x , G_y , G_z , G_α , sG_β , sG_γ) which are saved using the robot GUI interface.

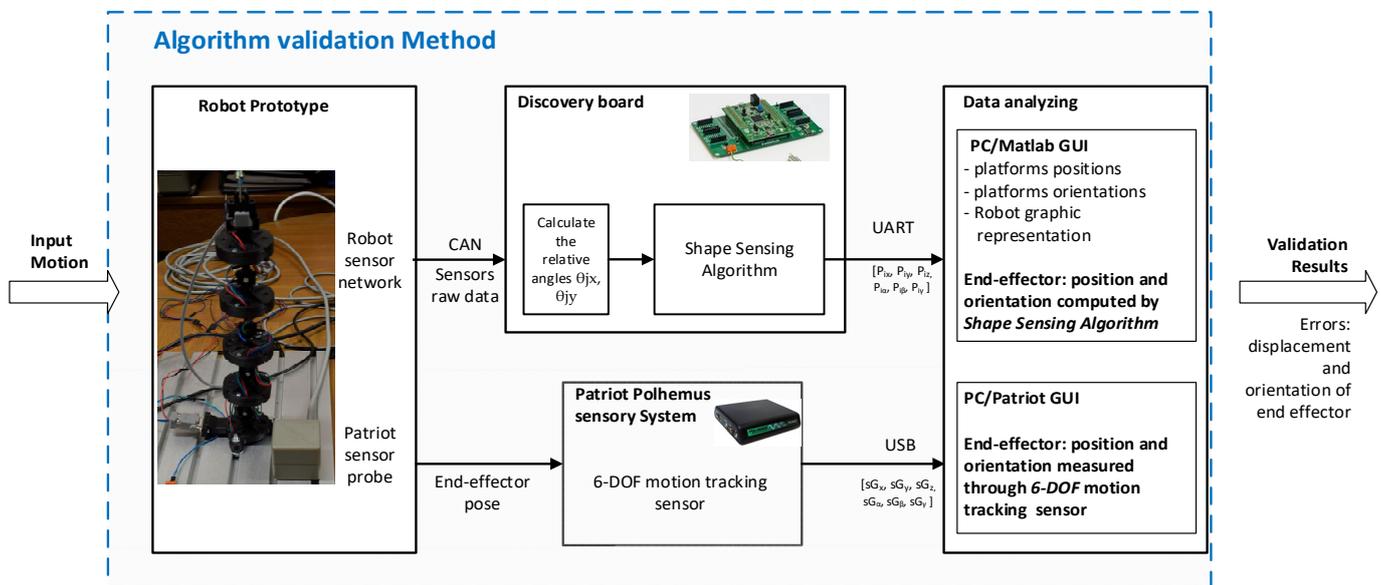


Figure 6. Conceptual diagram of the testing method for the end-effector pose.

4.2. Validation of Experimental Results

Using the experimental setup presented in Figure 5, a set of experiments were developed in order to measure and compare: relative angles between platforms, position and orientation of the platforms and trajectory of the end effector. In the experiments, an arbitrary continuous movement was manually imposed to the robot modules. The variations of the movements' amplitudes and platform orientations aimed at covering multiple regions of the robot workspace, thus assuring a set of representative data. The sensors' data were sent to dSPACE Platform and Discovery board as input data for calculating the robot shape (Figure 7).

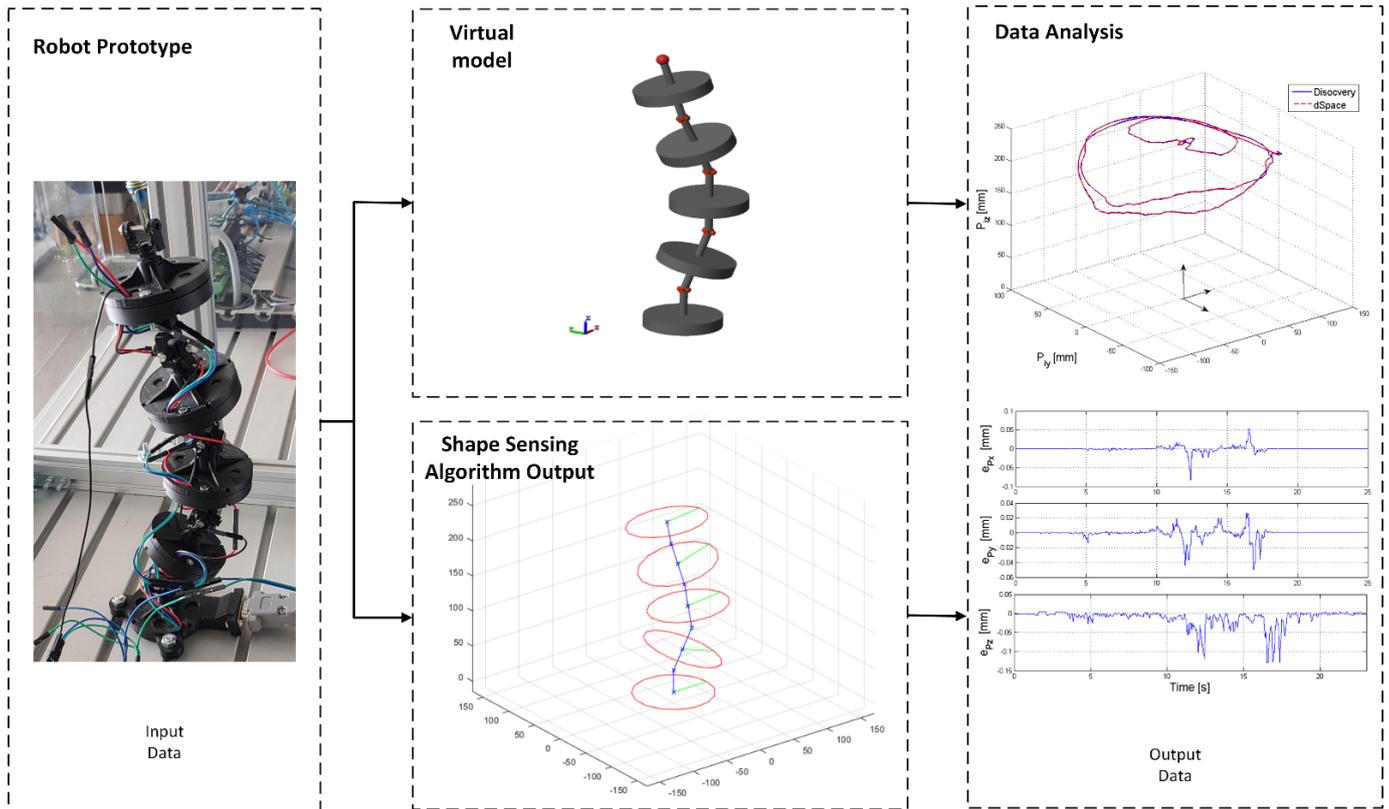


Figure 7. HIL testing method—validation of experimental results.

The experimental results are presented in Figures 8–11. In these figures, the values calculated using Discovery board are represented with continuous line and the values measured using dSPACE platform are represented with dash lines.

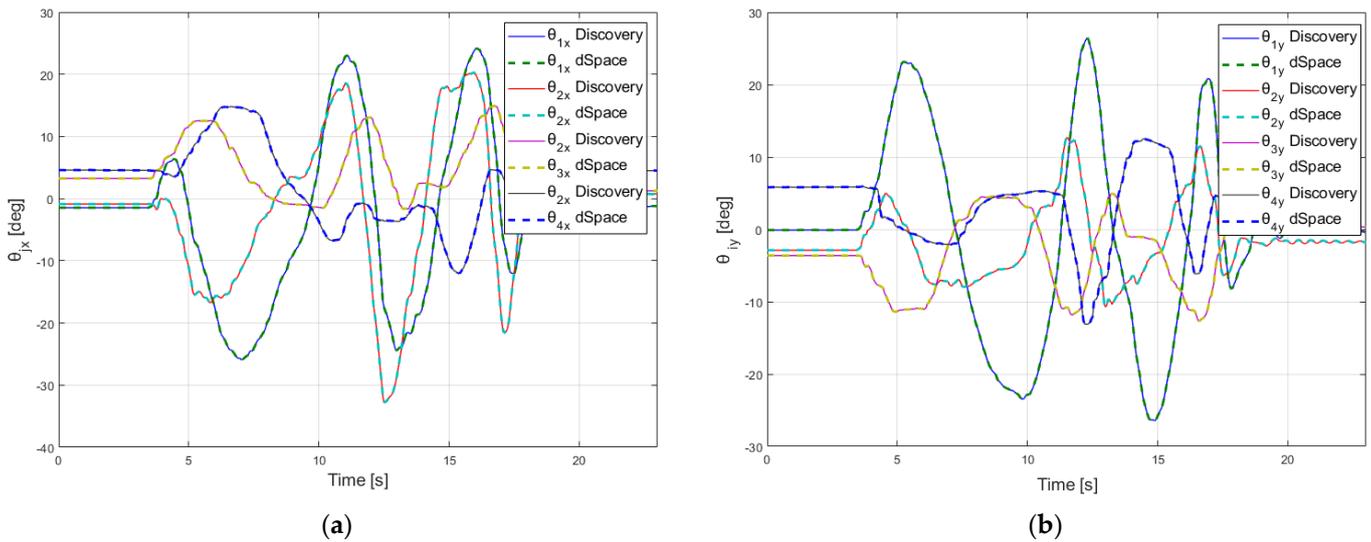


Figure 8. Variation of the relative angles between platforms (a) angle θ_{jx} (b) angle θ_{jy} .

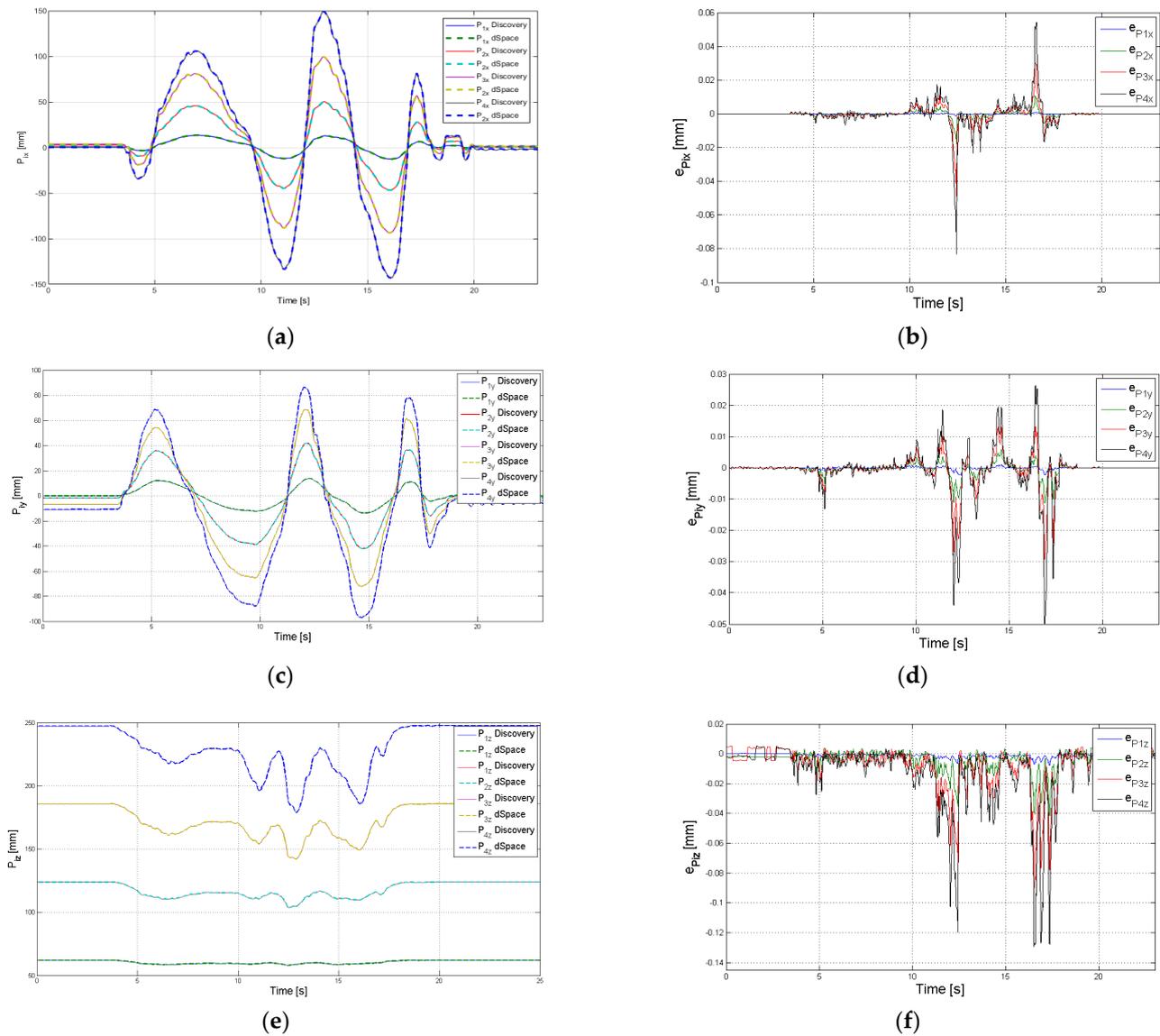


Figure 9. Experimental results for the position of each module of the robot: (a) parameters P_{jx} (b) absolute error of P_{jx} (c) parameters P_{jy} (d) absolute error of P_{jy} (e) parameters P_{jz} (f) absolute error of P_{jz} .

The raw data that are transmitted through the CAN sensor network, are used to calculate the relative angles between all the robot's consecutive platforms. In Figure 8, the variations of the angles θ_{jx} and θ_{jy} ($j = 0..4$) are presented for each module. The dSPACE and Discovery boards calculate these angles independently and the maximum of the absolute errors between these signals is negligible (less than 10^{-6} deg) being affected only by the sample time used by each of the two environments.

The variation of the position of each platform is presented in Figure 9 (parameter P_{jx} ($j = 1..4$) in Figure 9a, parameter P_{jy} ($j = 1..4$) in Figure 9c and parameter P_{jz} ($j = 1..4$) in Figure 9e. The absolute error for each platform for the three parameters is presented in Figure 9b,d,f. The obtained maximum absolute error for P_{jx} parameter is 0.083 mm, for P_{jy} is 0.049 mm and for P_{jz} parameter is 0.129 mm. As can be seen, the absolute errors are higher in the upper platforms, which is caused by a much bigger amplitude of position variation for these modules in comparison with the modules at the robot base.

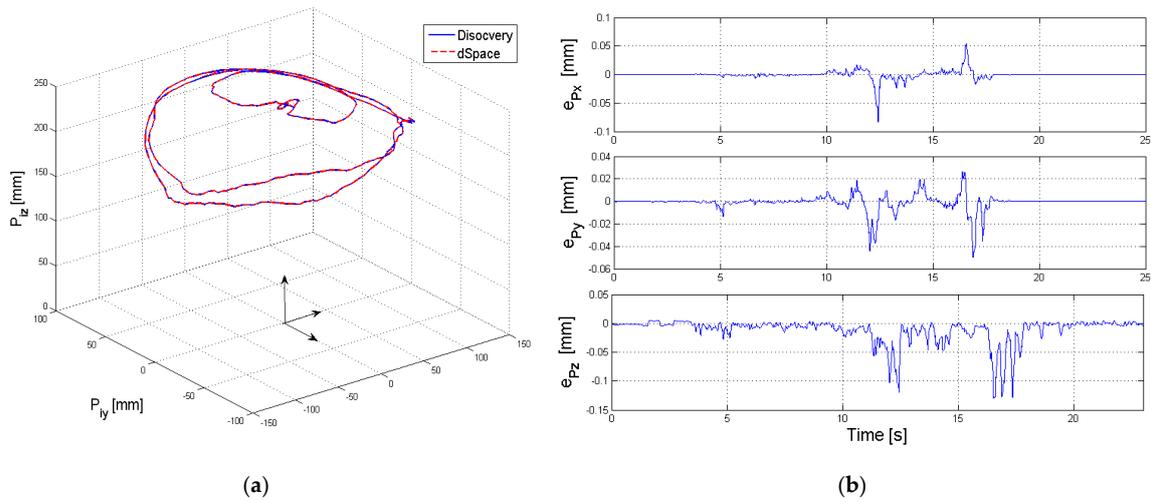


Figure 10. Trajectory of the end effector (4th module) (a) 3D representation (b) absolute positioning error along axis O_x , O_y and O_z for the 4th module.

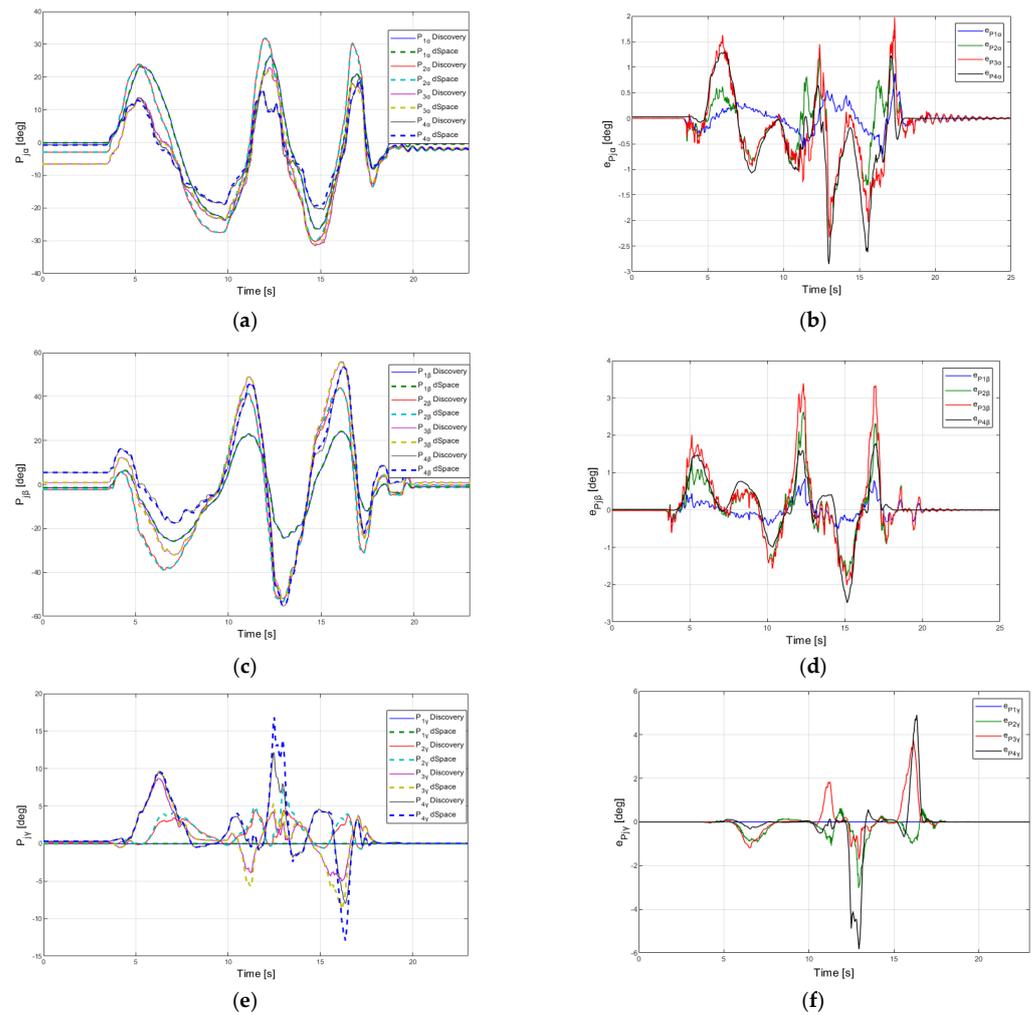


Figure 11. Experimental results for the orientation of each module of the robot for (a) parameter $P_{j\alpha}$ (b) absolute error of $P_{j\alpha}$ (c) parameter $P_{j\beta}$ (d) absolute error of $P_{j\beta}$ (e) parameter $P_{j\gamma}$ (f) absolute error of $P_{j\gamma}$.

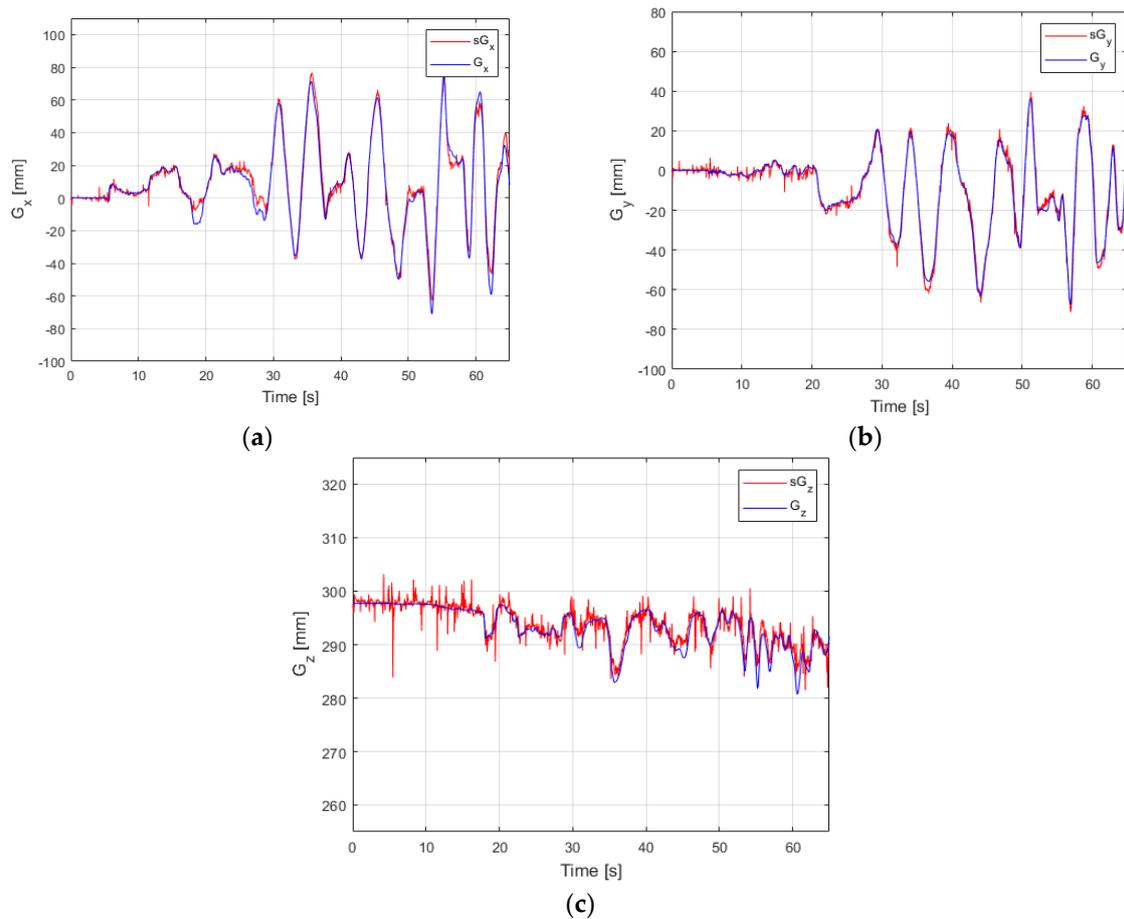


Figure 12. Experimental result—position of the end effector (a) parameter G_x (b) parameter G_y (c) parameter G_z .

The position parameters P_{jx}, P_{jy}, P_{jz} ($j = 1 \dots 4$) are used to reconstruct the robot shape. A 3D representation of the structure is represented in real time in the robot GUI interface that displays the data calculated by the Discovery board (Figure 4). The MAE errors for the fourth module were calculated, the obtained values are: for P_{4x} the MAE error is 0.0028 mm at a variation of 292.4 mm, for P_{4y} the MAE error is 0.003 mm at a variation of 183.5 mm and for P_{4z} the MAE error is 0.0116 mm at a variation of 68.9 mm.

The 3D representation of the end effector trajectory (position of the characteristic point for the 4th module $\{P_{4x}, P_{4y}, P_{4z}\}$) is presented in Figure 10a.

The variations of the platform's orientations are presented in Figure 11 (parameter $P_{j\alpha}$ ($j = 1 \dots 4$) in Figure 11a, parameter $P_{j\beta}$ ($j = 1 \dots 4$) in Figure 11c and parameter $P_{j\gamma}$ ($j = 1 \dots 4$) in Figure 11d). The MAE errors for parameter $P_{j\alpha}$ (Figure 11b,d,e) are 0.404 deg at a variation of 40.4 deg, for parameter $P_{j\beta}$ is 0.365 deg for a variation of 109.1 deg and for parameter $P_{j\gamma}$ is 0.291 deg for a variation of 20.1 deg.

After the analysis of the results, for this set of experimental data, it was observed that MAE errors for position varied between 0.0028 and 0.0116 mm and for orientation between 0.291 and 0.404 deg for the fourth module of the robot. These values were directly affected by the amplitude of the movements of these parameters and the hardware characteristics of the two computational systems (different discretization, different calculation precision etc.).

By using the second testing method (Figure 6), a set of experiments was developed to validate shape sensing computational system by measuring the end-effector pose. The experimental data results used in the validation process are presented in Figures 12 and 13. In these figures, the data measured by using the Patriot sensor are represented with a red line and the data calculated by using the shape sensing computational system proposed in the article with a blue line.

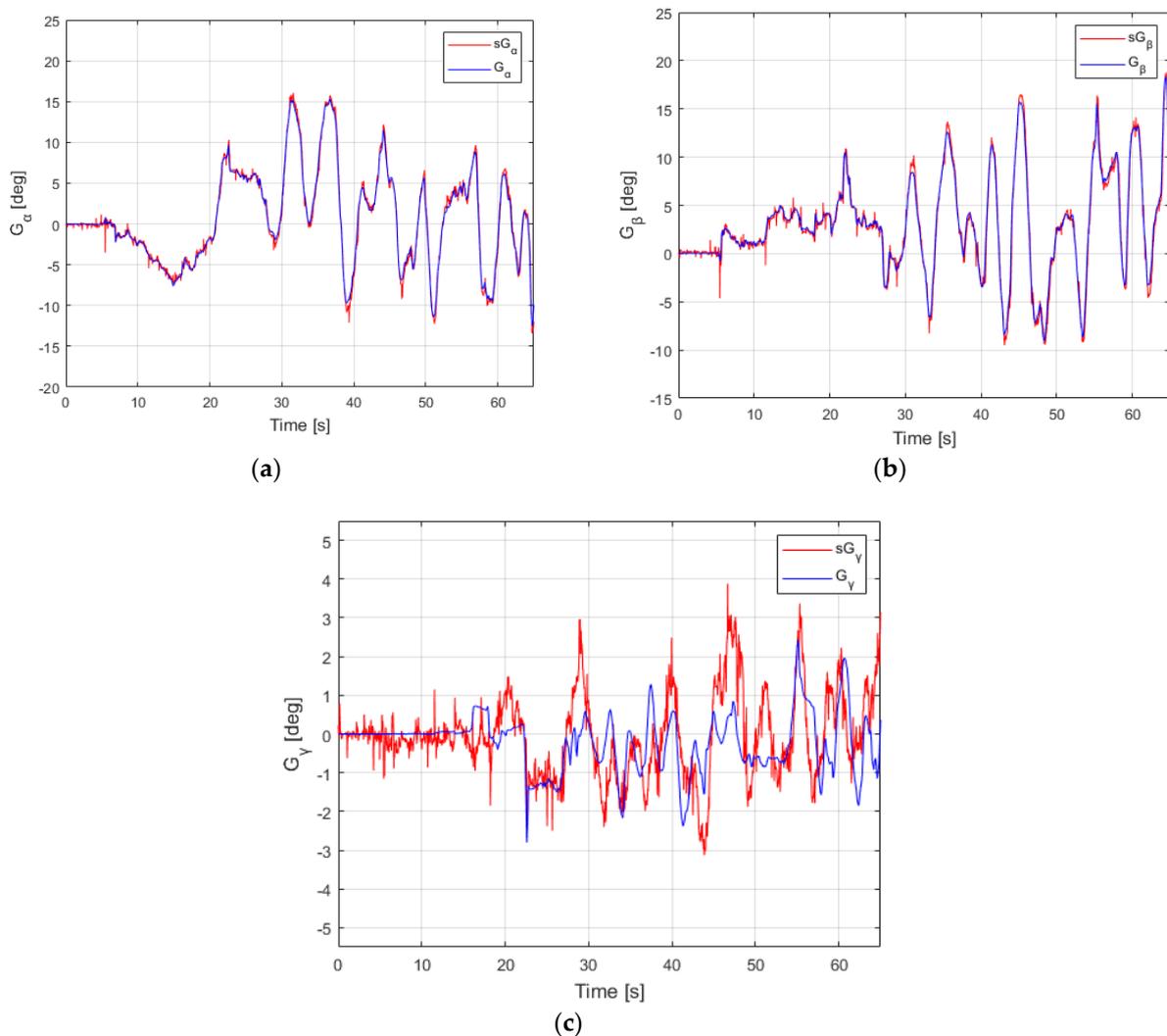


Figure 13. Experimental results—orientation of the end effector (a) parameter G_α (b) parameter G_β (c) parameter G_γ .

The MAE error obtained for parameter G_x is 2.91 mm at a variation of 147.4 mm, for parameter G_y is 1.52 at a variation of 103.3 mm and for parameter G_z the error is 0.539 mm for a range of 17.4 mm.

The MAE error obtained for parameter G_α is 0.33 deg at a range of 27.8 deg, for parameter G_β is 0.36 at a range of 27.4 deg and for parameter G_γ is 0.36 at a range of 5.2 deg.

During the experiments, it was observed that the orientation sensing along O_z axis for this type of sensors was influenced by the presence of electromagnetic waves. In this context, design restrictions are imposed concerning the electronic/magnetic components that are not allowed to be present in the closed proximity of the sensor.

An advantage of using this type of sensors is that the cumulative errors are reduced, due to the fact that the orientation between the platforms of each module is calculated independently based on the sensor data that are absolute values.

Analyzing the numerical results obtained with the two testing methods it was observed the follows. Using the HIL testing method, the MAE errors obtained for positioning were less than 0.0116 mm and for orientation were less than 0.404 deg when the robot shape was evaluated. Related to evaluation of the end effector pose, using the second testing method, it was observed that the MAE errors were less than 2.91 mm for X, Y, Z position and less than 0.36 deg for orientation angles. Taking into account the obtained results (effectiveness

and performance) it can be concluded that the shape sensing computational system can be used as part of the control system of the Python robot.

5. Conclusions

The paper presented a new approach for shape sensing of a hyper redundant robot with articulated joint-based rigid structure. A shape sensing computational system (which integrates a dedicated network of AHRS IMU sensors and a shape sensing algorithm) was proposed and developed and two testing methods were implemented for validating its effectiveness and performance.

The proposed computational system offers several advantages: reduced computational time, which makes the algorithm more feasible for real time computation; reduced cumulative errors due to the absolute measurements of the sensor network, improved robot design related to the system dimensions and communication buses.

The proposed HIL testing method uses a virtual robot which runs on a real time HIL simulation platform (dSPACE), in parallel with the computational system. The experimental data outputs were compared and the results validate the proposed algorithm. In the experiments developed, the MAE errors for positioning were less than 0.0116 mm and for orientation were less than 0.404 deg. The results obtained offer a good perspective for using the proposed sensor network in the development of the robot with application in harvesting and inspection tasks.

The second testing method was used to capture the pose of the end-effector with a state-of-the-art motion capture system and then it was compared with the one provided by the shape sensing computational system proposed in this paper. Experimental results were analyzed and MAE errors were less than 2.91 mm for X,Y,Z position and less than 0.36 deg for orientation angles of the end-effector.

Both testing methods validated the proposed shape sensing computational system, which is further used to implement the Python robot control system. Other solutions to improve the shape sensing computational system performance could be addressed also. The authors intend to use flexible soft sensors from Bend Labs and to fuse the data received from the two sensor systems. This would allow an increase in precision and potential new application in domains where high accuracy is needed.

Author Contributions: Conceptualization, O.H. and C.L.; methodology, C.L. and O.H.; software, C.L.; validation, C.L., C.R. and O.H.; formal analysis, O.H.; investigation, C.R.; resources, C.L.; data curation, C.L.; writing—original draft preparation, C.L. and C.R.; writing—review and editing, C.L., C.R. and O.H.; visualization, C.L.; supervision, O.H.; project administration, O.H.; funding acquisition, O.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Romanian Ministry of Education and Research, CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2019-4939, within PNCDI III <https://uefiscdi.gov.ro/proiect-experimental-demonstrativ-ped> (accessed on 22 December 2021).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by a grant of the Romanian Ministry of Education and Research, CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2019-4939, within PNCDI III.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Martín-Barrio, A. Design, Modelling, Control and Teleoperation of Hyper-Redundant Robots. Ph.D. Thesis, Universidad Politécnica de Madrid, Madrid, Spain, 2020. Available online: https://oa.upm.es/65161/1/ANDRES_MARTIN_BARRIO.pdf (accessed on 10 November 2021).
2. Hirose, S.; Mori, M. Biologically Inspired Snake-like Robots. In Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, Shenyang, China, 22–26 August 2004; pp. 1–7. [\[CrossRef\]](#)
3. Liu, J.; Tong, Y.; Liu, J. Review of snake robots in constrained environments. *Robot. Auton. Syst.* **2021**, *101*, 103785. [\[CrossRef\]](#)
4. Robinson, G.; Davies, J.B.C. Continuum Robots—A State of the Art. In Proceedings of the 1999 IEEE International Conference on Robotics & Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 4, pp. 2849–2854. [\[CrossRef\]](#)
5. Kolachalama, S.; Lakshmanan, S. Continuum Robots for Manipulation Applications: A Survey. *J. Robot.* **2020**, *2020*, 4187048. [\[CrossRef\]](#)
6. Series II, X125 System, Datasheet. Available online: <https://www.ocrobotics.com/technology-/series-ii-x125-system> (accessed on 10 November 2021).
7. Martín-Barrio, A.; Roldán-Gómez, J.J.; Rodríguez, I.; Cerro, J.; Barrientos, A. Design of a Hyper-Redundant Robot and Teleoperation Using Mixed Reality for Inspection Tasks. *Sensors* **2020**, *20*, 2181. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Gonzalez-de-Santos, P.; Fernández, R.; Sepúlveda, D.; Navas, E.; Emmi, L.; Armada, M. Field Robots for Intelligent Farms—Inhering Features from Industry. *Agronomy* **2020**, *10*, 1638. [\[CrossRef\]](#)
9. Navas, E.; Fernández, R.; Sepúlveda, D.; Armada, M.; Gonzalez-de-Santos, P. Soft Grippers for Automatic Crop Harvesting: A Review. *Sensors* **2021**, *21*, 2689. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Rad, C.; Hancu, O.; Lapusan, C. Aspects regarding “soft” grasping in smart agricultural harvesting tasks. *Acta Tech. Napoc. Ser. Appl. Math. Mech. Eng.* **2020**, *63*, 389–394.
11. Chowdhary, G.; Gazzola, M.; Krishnan, G.; Soman, C.; Lovell, S. Soft Robotics as an Enabling Technology for Agroforestry Practice and Research. *Sustainability* **2019**, *11*, 6751. [\[CrossRef\]](#)
12. Chen, L.; Ma, Y.; Zhang, Y.; Liu, J. Obstacle Avoidance and Multitarget Tracking of a Super Redundant Modular Manipulator Based on Bezier Curve and Particle Swarm Optimization. *Chin. J. Mech. Eng.* **2020**, *33*, 71. [\[CrossRef\]](#)
13. Zhao, Y.; Jin, L.; Zhang, P.; Li, J. Inverse Displacement Analysis of a Hyper-redundant Elephant’s Trunk Robot. *J. Bionic Eng.* **2018**, *15*, 397–407. [\[CrossRef\]](#)
14. Martín, A.; Barrientos, A.; del Cerro, J. The Natural-CCD Algorithm, a Novel Method to Solve the Inverse Kinematics of Hyper-redundant and Soft Robots. *Soft Robot.* **2018**, *5*, 242–257. [\[CrossRef\]](#)
15. Behrens, R.; Kuchler, C.; Förster, T.; Elkmann, N. Kinematics analysis of a 3-DOF joint for a novel hyper-redundant robot arm. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3224–3229. [\[CrossRef\]](#)
16. Floris, I.; Adam, J.M.; Calderón, P.A.; Sales, S. Fiber Optic Shape Sensors: A comprehensive review. *Opt. Lasers Eng.* **2021**, *139*, 106508. [\[CrossRef\]](#)
17. Amanzadeh, M.; Aminossadati, S.M.; Kizil, M.S.; Rakić, A.D. Recent developments in fibre optic shape sensing. *Measurement* **2018**, *128*, 119–137. [\[CrossRef\]](#)
18. Roan, P.; Deshpande, N.; Wang, Y.; Pitzer, B. Manipulator State Estimation with Low Cost Accelerometers and Gyroscopes. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 4822–4827. [\[CrossRef\]](#)
19. Adnan, N.H.; Wan, K.; Shahriman, A.; Za’ba, S.K.; Desa, H.; Aziz, M.A.A. The development of a low cost data glove by using flexible bend sensor for resistive interfaces. In Proceedings of the 2012 2nd International Malaysia-Ireland Joint Symposium on Engineering, Science and Business (IMiEJS2012), Kuala Lumpur, Malaysia, 18–20 June 2012. pp. 579–587.
20. Bendlabs. Flex Sensors by Bend Labs. 2021. Available online: <https://www.bendlabs.com/> (accessed on 10 November 2021).
21. Koh, J.H.B.; Jeong, T.; Han, S.; Li, W.; Rhode, K.; Noh, Y. Optoelectronic Sensor-based Shape Sensing Approach for Flexible Manipulators. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Berlin, Germany, 23–27 July 2019; pp. 3199–3203. [\[CrossRef\]](#)
22. Dementyev, A.; Kao, C.H.-L.; Paradiso, J.A. SensorTape: Modular and Programmable 3D-Aware Dense Sensor Network on a Tape. In Proceedings of the 2015 28th Annual ACM Symposium, Charlotte, NC, USA, 8–11 November 2015. [\[CrossRef\]](#)
23. Schmitz, A.; Thompson, A.J.; Berthet-Rayne, P.; Seneci, C.A.; Wisanuvej, P.; Yang, G.-Z. Shape Sensing of Miniature Snake-Like Robots Using Optical Fibers. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 947–952. [\[CrossRef\]](#)
24. Quigley, M.; Brewer, R.; Soundararaj, S.P.; Pradeep, V.; Le, Q.; Ng, A.Y. Low-cost accelerometers for robotic manipulator perception. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 6168–6174. [\[CrossRef\]](#)
25. Wright, C.; Buchan, A.; Brown, B.; Geist, J.; Schwerin, M.; Rollinson, D.; Tesch, M.; Choset, H. Design and architecture of the unified modular snake robot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 4347–4354. [\[CrossRef\]](#)
26. Cheng, P.; Oelmann, B. Joint-Angle Measurement Using Accelerometers and Gyroscopes—A Survey. *IEEE Trans. Instrum. Meas.* **2009**, *59*, 404–414. [\[CrossRef\]](#)

27. Navidi, N.; Landry, R. A New Perspective on Low-Cost MEMS-Based AHRS Determination. *Sensors* **2021**, *21*, 1383. [[CrossRef](#)]
28. Zhao, J. A Review of Wearable IMU (Inertial-Measurement-Unit)-based Pose Estimation and Drift Reduction Technologies. *IOP Conf. Ser. J. Phys. Conf. Ser.* **2018**, *1087*, 042003. [[CrossRef](#)]
29. Justa, J.; Šmídl, V.; Hamáček, A. Fast AHRS Filter for Accelerometer, Magnetometer, and Gyroscope Combination with Separated Sensor Corrections. *Sensors* **2020**, *20*, 3824. [[CrossRef](#)]
30. Seel, T.; Kok, M.; McGinnie, R.S. Inertial Sensors—Applications and Challenges in a Nutshell. *Sensors* **2020**, *20*, 6221. [[CrossRef](#)]
31. Bancroft, J.B.; Lachapelle, G. Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors* **2011**, *11*, 6771–6798. [[CrossRef](#)]
32. Wittmann, F.; Lambercy, O.; Gassert, R. Magnetometer-based drift correction during rest in IMU arm motion tracking. *Sensors* **2019**, *19*, 1312. [[CrossRef](#)]
33. WitMotion WT901, Datasheet. Available online: <https://www.wit-motion.com/gyroscope-module/Witmotion-wt901-ttl-i2c.html> (accessed on 10 November 2021).
34. Clemente, F.M.; Akyildiz, Z.; Pino-Ortega, J.; Rico-González, M. Validity and Reliability of the Inertial Measurement Unit for Barbell Velocity Assessments: A Systematic Review. *Sensors* **2021**, *21*, 2511. [[CrossRef](#)]
35. Zhou, L.; Fischer, E.; Tunca, C.; Brahms, C.M.; Ersoy, C.; Granacher, U.; Arnrich, B. How We Found Our IMU: Guidelines to IMU Selection and a Comparison of Seven IMUs for Pervasive Healthcare Applications. *Sensors* **2020**, *20*, 4090. [[CrossRef](#)] [[PubMed](#)]
36. Du, G.; Zhang, P. IMU-Based Online Kinematic Calibration of Robot Manipulator. *Sci. World J.* **2013**, *2013*, 139738. [[CrossRef](#)] [[PubMed](#)]
37. Coviello, G.; Avitabile, G.; Florio, A.; Talarico, C.; Wang-Roveda, J.M. A Novel Low-Power Time Synchronization Algorithm Based on a Fractional Approach for Wireless Body Area Networks. *IEEE Access* **2021**, *9*, 134916–134928. [[CrossRef](#)]
38. Kang, B.S.; Park, E.J. Modeling and Control of an Intrinsic Continuum Robot Actuated by Pneumatic Artificial Muscles. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016; pp. 1157–1162. [[CrossRef](#)]
39. Peng, Y.; Liu, Y.; Yang, Y.; Liu, N.; Sun, Y.; Liu, Y.; Luo, J. Development of continuum manipulator actuated by thin McKibben pneumatic artificial muscle. *Mechatronics* **2019**, *60*, 56–65. [[CrossRef](#)]
40. Zhang, Z.; Shang, J.; Seneci, C.; Yang, G.Z. Snake Robot Shape Sensing Using Micro-inertial Sensors. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 831–836. [[CrossRef](#)]
41. Luo, M.; Li, E.; Guo, R.; Liu, X.; Liang, Z. End-Effector Pose Estimation in Complex Environments Using Complementary Enhancement and Adaptive Fusion of Multisensor. *J. Sens.* **2021**, *2021*, 5550850. [[CrossRef](#)]
42. Zhao, X.; Dou, L.; Su, Z.; Liu, N. Study of the Navigation Method for a Snake Robot Based on the Kinematics Model with MEMS IMU. *Sensors* **2018**, *18*, 879. [[CrossRef](#)] [[PubMed](#)]
43. Lapusan, C.; Rad, C.; Hancu, O. Kinematic analysis of a hyper-redundant robot with application in vertical farming. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1190*, 012014. [[CrossRef](#)]
44. Tang, L.; Wang, J.; Zheng, Y.; Gu, G.; Zhu, L.; Zhu, X. Design of a cable-driven hyper-redundant robot with experimental validation. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417734458. [[CrossRef](#)]
45. Lapusan, C.; Hancu, O.; Rad, C. Quaternion-Based Approach for Solving the Direct Kinematics of a Modular Hyper Redundant Robot. *Acta Tech. Napoc. Ser. Appl. Math. Mech. Eng.* **2020**, *63*, 363–366.
46. Fathy, H.K.; Filipi, Z.S.; Hagen, J.; Stein, J.L. Review of hardware-in-the-loop simulation and its prospects in the automotive area. In Proceedings of the 2006 Defense and Security Symposium, Orlando, FL, USA, 17–21 April 2006; Volume 6228, p. 62280. [[CrossRef](#)]
47. Polhemus, Patriot 6DOF Tracker Data Sheet. Available online: https://polhemus.com/_assets/img/PATRIOT_brochure.pdf (accessed on 22 December 2021).