HeRo: An Open Platform for Robotics Research and Education

Paulo A. F. Rezeck[1] and Hector Azpurua[2] and Luiz Chaimowicz[3]

***Abstract*— In this paper, we present a novel platform for swarm robotics that is low cost, easy to assemble using off the shelf components and is deeply integrated with the most used robotic framework available today: ROS (Robot Operating System). The robotic platform is completely open, composed by a 3D printed body and open source software. We describe its architecture, present its main features and evaluate its functionalities executing real experiments using a couple of robots. We concluded that the proposed swarm platform is a feasible and usable system for education and research, given its reduced cost, easy of use and small size.**

I. INTRODUCTION

Small, simple, yet capable, robots have generated attention of the research community towards practical applications of swarm robotics. Swarm robotics is a relatively new research area inspired from biological systems such as ant or bee colonies. It is characterized by simplicity of individuals, local sensing and communication capabilities, robustness, scalability, parallelism in task execution, flexibility, and decentralized control [1].

Groups of simpler robots working as a swarm can produce complex collective behaviors for solving tasks such as: aggregation [2], collective transportation [3], collective search and exploration [4], foraging [5], assembling and construction [6], pattern formation [7] among many others.

Most of the experiments using large groups of robots are performed in simulation, due to the difficulties in performing real experiments with swarms and the high cost of small robotic platforms. Unfortunately, simulations do not always capture all the complexities and results of real experimentation. But, in recent years, different low-cost mobile robotic platforms have been developed, bridging this gap between simulations in swarm robotics and the experiments with real robots.

Since these different platforms have been introduced, many researches have successfully employed them in various research applications. However, this trend has not affected all researchers, mainly due to the cost of platforms and components and the logistics for acquiring them.

Believing that the Maker Movement and DIY (Do It Yourself) trends offers many new opportunities for robotics researchers and educators to produce highly capable mobile robots at lower costs, in this work we present the design of HeRo: a significant low-cost platform for about 18 USD, that can be used for robotic swarm applications and education efforts. A annotated picture of HeRo is shown in Figure 1.

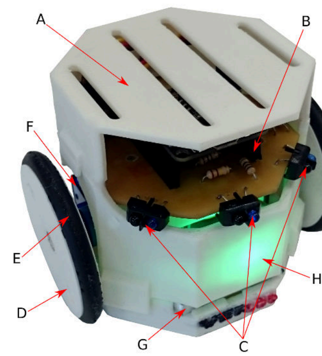


Fig. 1: Hero platform. Some features are: (A) 3D printer body, (B) Circuit board, (C) IR sensors, (D) Wheel, (E) Rubber O-ring, (F) Servo Motor, (G) Battery and (H) RGB LED.

Hero is affordable, open-platform, 3D printable and easy to be assemble with off the shelf components. Also, it is deeply integrated with ROS (Robot Operating System) [8], one of the most popular open-source robot operating system. HeRo's main objective is to be cost effective, so we removed all sensors that were not fully necessary for swarm simulations: for example, on-board cameras and inertial sensors are not integrated on the platform. As we use an external tracking system with cameras, we only need proximity sensors (we use 3) to perform basic autonomous decisions. The software platform is developed using ROS with common tools and following strict standards to be easy integrable with other types of projects or developments.

The rest of this paper is organized as follows: in Section II, we give an overview of related works regarding small robot platforms and systems. In Section III, we describe the hardware architecture of HeRo, followed by the design of the software stack in Section IV. The experiments and results are described in Section V, and, finally, the conclusions and directions for future work are discussed in Section VI.

II. RELATED WORK

A wide range of small and relatively simple robots have been proposed and some of them are available on the market. In this section we survey a subset of them that we think to be relevant for swarm experimentation. Table I shows a comparison among them.

Colias [9] is a robotic platform developed at the University of Lincoln, UK for swarm robotic applications. It is an

[1]Paulo A. F. Rezeck is pursuing a MsC in Computer Science at the Universidade Federal de Minas Gerais, Brazil rezeck@dcc.ufmg.br
[2]Hector Azpurua is pursuing a PhD in Computer Science at the Universidade Federal de Minas Gerais, Brazil hector.azpurua@dcc.ufmg.br
[3]Luiz Chaimowicz is an associate professor in the Department of Computer Science, Universidade Federal de Minas Gerais, Brazil chaimo@dcc.ufmg.br

978-1-5386-0956-9/17/$31.00 ©2017

TABLE I: Comparison of some swarm robotics platforms.

Robot	Cost	Sensor	Communication	Motion/Speed	Size	Autonomy
HeRo	**18 USD**	**distance**	**WiFi**	**wheel, 25 cm/s**	**8 cm**	**2-4 h**
Colias [9]	40 USD	distance, light, bump, bearing, range	IR	wheel, 35 cm/s	4 cm	1-3 h
Zooids [10]	50 USD	touch sensor	RF	wheel, 44 cm/s	2.6 cm	1-2 h
AMiR [11]	85 USD	distance, light, bearing	IR	wheel, 10 cm/s	6.5 cm	2 h
MicroMVP [12]	90 USD	None	Zigbee	wheel, N/A	8 cm	1-2 h
Kilobot [13]	100 USD	distance, light	IR	vibration, 1 cm/s	3.3 cm	3-24 h
Jasmine [14]	105 USD	distance, light, bearing	IR	wheel, N/A	3 cm	1-2 h
E-puck [15]	780 USD	distance, camera, bearing, acc, mic	Bluetooth	wheel, 13 cm/s	7.5 cm	1-10 h

open platform and costs about 40 USD. Colias is based on an 8-bit ATmega168 microcontroller and is equipped with IR sensors which provide proximity measurements and communication means with other robots. Colias comes with a set of basic software libraries for sensor reading and motion control. The programming is provided by the microcontroller's development environment and thus requires some knowledge of this kind of programming.

AMiR [11] is an older alternative to Colias. It is also an open platform and costs about 85 USD. This robot uses an 8-bit ATmega168 microcontroller and is equipped with a number of IR proximity sensors that are also used for communication and LEDs for status monitoring. The robot's microcontroller can be programmed from a PC by a dedicated programming interface. This robot has been successfully simulated in Player/Stage and has been used by various researchers and robotics educators.

Jasmine [14] is a robot platform designed for swarm applications, which is available for approximately 105 USD. It is an open platform with simulation capabilities. The basic version of the robot comes with an 8-bit ATmega168 main microcontroller and uses a number of IR sensors for proximity sensing, communication with other robots and light measurements, and LEDs for status monitoring. The capabilities of the robot can be extended by a number of customised boards including improved sensing, connectivity, etc.

The E-puck [15] is one of the most successful small size commercial robots and is used for both research and education. It hosts an variety of sensors including microphone arrays, proximity sensors, camera, accelerometers, and so on. The robot's microcontroller can be programmed through a serial or Bluetooth interface and the Bluetooth is also used for communication. However, the commercialized version of the basic e-puck costs about 780 USD and an extra 400 USD is needed to obtain an additional range and bearing module.

The Kilobot [13] was specially developed for swarm applications. It is an open-source platform and it is the winner of the AFRON robot design challenge with parts costing only 14 USD. However, it is now produced and distributed as a commercial product at a price of 100 USD. The robot has an 8-bit ATmega328 microcontroller and is equipped with ambient light and IR sensors for proximity readings and communication. The robot has an alternative moving principle based on vibration motors which requires a fairly smooth surface and results in a relatively slow movement. The robot's microcontroller can be programmed through a serial interface requiring a dedicated programming device. The robot has simulation support through the V-REP simulator.

Zooids [10] is a small robot platform designed for swarm applications available at approximate cost of 50 USD. The motors are placed non-colinearly to minimize the diameter at 2.6 cm. Even though the motors do not rotate around the same axis, the robot has the same net force and moment as would a robot with colinear motors. A 48MHz ARM microcontroller manages the overall logic computation and communicates wirelessly with the main master computer using a 2.4GHz radio chip.

Finally, MicroMVP [12] is a capable and affordable mobile robot platform with a small footprint. It has an open source design, utilizing the latest 3D printing technology and is extremely simple and robust. Significant care is also taken to ensure that only readily available components are used in this robot, which makes MICRO MVP a truly readily reproducible mobile robot platform.

The main contributions of the proposed platform compared to its counterparts is its low cost, its simplicity in terms of assembly, and its seamless integration with ROS. In the next sections we describe the hardware and software architecture of HeRo.

III. PLATFORM ARCHITECTURE AND DESIGN

Design decisions for our proposed robot were made based on three key factors: high modularity; maximum use of commercially available components for ease of production and assembly; and minimum possible price without sacrificing processing power and sensing capabilities.

To reach the design target, we explored different microcontroller families (Arduino Nano and NodeMcu), wireless technologies (Wifi, Bluetooth, and Radio Frequency), actuators (DC Motors, Stepper Motor and Servo Motors) and model designs for 3D printing.

By exploring this large set of possibilities, we eventually defined our architecture, shown in Figure 2, using NodeMcu for communication and control, and Servo Motors for mobility. We did not choose Arduinos, which is a trend in recent years, due to the limited processing power of the Arduino, which would cripple a fast IEEE 802.11 connection. The use of servos give us a good trade-off between size, control and speed, as other types of motors required extra components to work, making them difficult to integrate in a small body.

In the next sections we detail the various systems within HeRo.

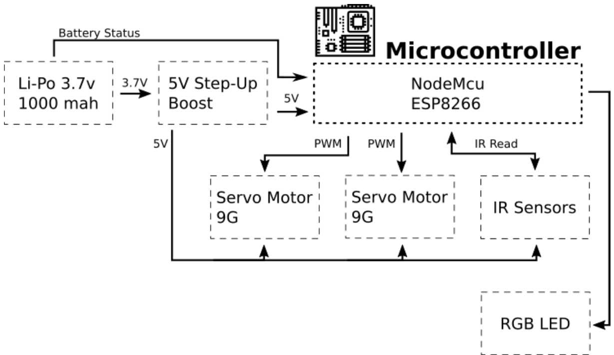


Fig. 2: Basic architecture of HeRo.

A. Motion

Two Micro Servo Motor SG90 (Figure 3b), adapted to continuous rotation, employ direct gears and two wheels with a diameter of 4.9 cm actuate HeRo with a maximum speed of 25 cm/s. To achieve good adherence we have coupled around the wheel a rubber o-ring (Figure 3f). The rotational speed for each motor is controlled individually using pulse-width modulation (PWM).

HeRo uses a differential-driven configuration, which is a simple method to control a mobile robot using a very basic motion control principle. The kinematic model, represented by linear and angular velocities $(\dot{x}, \dot{\phi})$ is given by:

$$\begin{aligned} \dot{x} &= \frac{r}{2}(u_r + u_l), \\ \dot{\phi} &= \frac{r}{L}(u_r - u_l). \end{aligned} \tag{1}$$

In this model, r is the wheel radius, u_r and u_l are the control inputs to the right and left wheels, respectively, and L is the distance between the two wheels.

Since each motor can behave differently, we defined a gain constant between the motors. This calibration process consists of recording the initial orientation, setting the robot to move forward in a straight line and comparing the final and initial orientation. We use the difference between the orientations to either increase or decrease the motor gain accordingly. This process is repeated until the initial and final orientations are sufficiently close.

Therefore, applying the inverse kinematic model for some desired linear and angular velocity, the respective control inputs can be computed by:

$$\begin{aligned} u_r &= \frac{2\dot{x} + \dot{\phi}L}{2r} * (1 - \delta), \\ u_l &= \frac{2\dot{x} - \dot{\phi}L}{2r} * (1 + \delta), \end{aligned} \tag{2}$$

where δ is the gain between the motors.

B. Controller and Communication

The controller, shown in Figure 3a, for the robot serves two functions. Firstly, it controls all the low-level electronics such as motors, sensors, and the RGB LED (used for displaying information to the operator). Secondly, it interfaces a TCP/IP communication with the ROS platform, running on the user's computer.

The controller is an ESP8266 microprocessor encapsulated into NodeMcu V3 board [16]. The ESP8266 is a low-cost IEEE 802.11 chip with full TCP/IP stack and MCU (microcontroller unit) capability, which runs at 80 Mhz (can be over-clocked to 160 Mhz) and has 128KB of memory, sufficient space for running the controller software. Two key features of this controller that we take advantage of are: 16 pulse width modulation (PWM) channels used for controlling the motors, RGB LED, and sensors; and 10-bit analog-to-digital converters used for measuring the incoming infrared light intensity from IR sensors.

Some of the previous works in swarm robots focused on direct robot-to-robot communication using generally low power, short range sensors such as IR communication. To turn the HeRo platform into a true networked robotic system, suitable for running experiments requiring local and global communication, we used embedded IEEE 802.11 communication because is low power and operates on the standard and robust TCP/IP protocol, which works perfectly with the ROS architecture.

C. Sensory System

The platform uses only IR proximity sensors to avoid obstacles as well collisions with other robots. The IR sensory system consists of three TCRT5000 long-range sensors (Figure 3c). Those sensors are placed in front of the robot and have a range of approximately 20 cm.

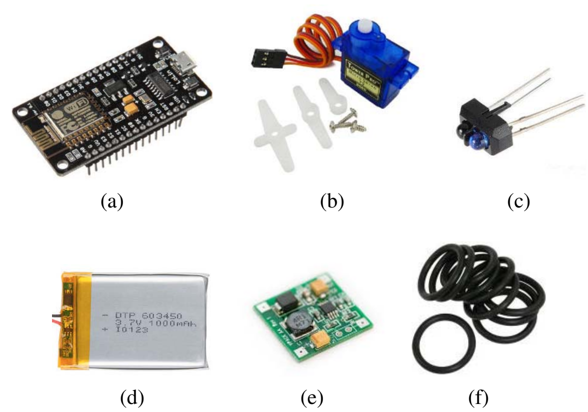


Fig. 3: Key components of HeRo. (a) NodeMcu v3 board with built-in Wi-Fi support. (b) Servo Motor SG90. (c) IR Sensor TCRT5000. (d) Lithium-ion battery 1000 mah. (e) 5V Step-Up boost. (f) Rubber O-ring used as wheel tires

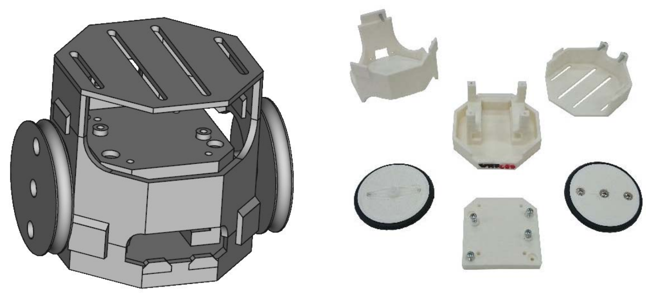
D. Power System

To power the entire robot, we use a a 3.7 V 1000 mAh Li-Po battery, shown in Figure 3d. This battery can power the whole system for 2-4 hours depending on the robot's activity level. Connected to this battery, there is a 5 V step-up boost, shown in Figure 3e, that correctly regulates the voltage to supply the overall system.

E. 3D Printer Parts

Extensive work has been devoted recently to the development of 3D printing or additive layer manufacturing technologies, as well their applications in robotics. The use of 3D printing is essential in the design phase, allowing both the precise fitting of the components and a quick screw-less assembly.

In order to be compatible with common 3D printing technology, the parts of HeRo were designed to be modular and at the same time structured to have minimal number of small features and parts. HeRo consists of 5 main 3D printer parts: main frame, board support, wheel, middle case and upper case as seen on Figure 4.



(a) (b)

Fig. 4: HeRo platform parts: (a) 3D Model developed in FreeCAD and (b) 3D printed parts.

F. Camera Platform for Robot Tracking

Besides the robot itself, we also built a visual-based global localization system (Figure 5) composed by one external camera (we selected the Logitech Pro 9000, 1600 x 1200 max resolution) that tracks the pose of each robot. To facilitate the tracking, each robot is outfitted with a 2D, monochromatic bar code. Each bar code is tracked with the help of a ROS package wrapped for Alvar, an open source AR tag tracking library.

This package features adaptive thresholding to handle a variety of lighting conditions, optical flow based tracking for more stable pose estimation, and an improved tag identification method that does not significantly slow down as the number of tags increases.

G. Cost

Table II gives a summary of the cost of some components used in HeRo. All part prices are assuming purchase of one robot using common part distributors from the Internet. We expect this cost would be greatly reduced if parts are acquired in bulk directly from manufacturers.

IV. SOFTWARE STACK

In order to provide for simple programming and user-friendly robotic implementation, we provide HeRo as a ROS compatible robot by connecting them using a TCP/IP connection as shown in the diagram in Figure 6. All necessary

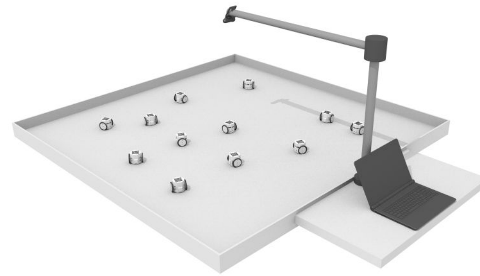


Fig. 5: Platform configuration for swarm experiments. Several HeRo robots are controlled by a system composed of a computer running ROS and one or more cameras for tracking.

TABLE II: Parts cost per robot.

Components	Cost (USD)
2 Servo Motors SG90	3.00
NodeMCU v3	3.50
Li-Po 3.7v 1000 mah	4.50
Step Up Boost 5V	0.60
Circuit Board and 3D Printer Parts	4.00
Various Electronics	2.30
Total	17.90

material and documentation for implementing HeRo was placed on Github.

A. Firmware

To implement the firmware we have used the Arduino IDE, since it is commonly used and allow us to easily build an interface between ROS and the sensors and actuators.

Into the firmware, we can set the communication parameters of the robot (e.g essid, password and ROS master IP) as well as their intrinsic parameters (e.g. robot id, motor calibrations and gain between the motors.)

B. ROS Package

We have also developed a ROS package, which runs on a external computer and provides direct communication with the robots using ROS serial node in TCP mode. This node interfaces a pre-configured TCP port in the ROS master to reply all the topics and the subscriber created in the robot microcontroller. Once this node is running, the user can access all functionalities of all robots at a rate of 100 Mhz.

Other functionalities of this package consist of: (i) a teleoperation node to manually control the robot (ii) an AR tag tracking to provide a global position of the robots, and (iii) a potential field controller to move the robot autonomously.

V. EXPERIMENTS

To show some of HeRo's capabilities, we performed some experiments using two robots in different scenarios. Basically, we performed four different experiments using both a closed loop control based on visual information from the external camera and also using only local data from the IR sensors.

In the first set of experiments (shown in Figure 7), we wanted to evaluate the feasibility of complex tasks using

https://github.com/rezeck/hero_driver
Videos of the experiments can be found at https://youtu.be/foQDcUG9Arg

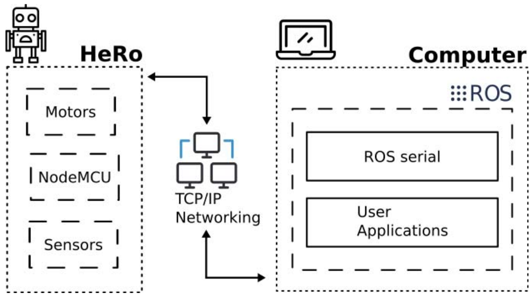
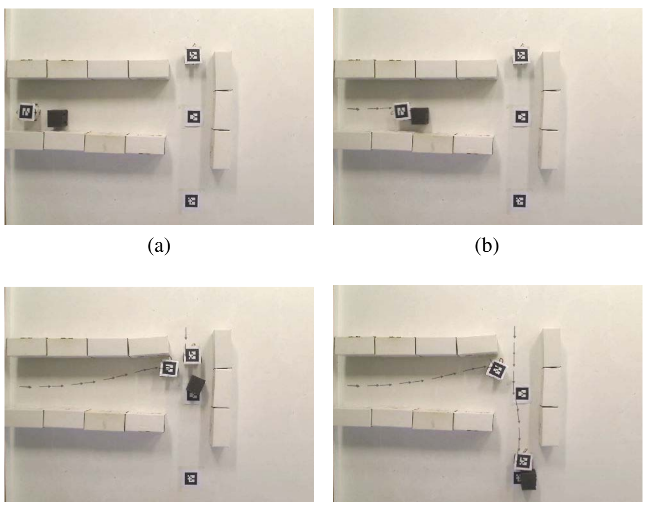


Fig. 6: Software Stack. HeRo robot connects by TCP to ROS to share control and sensor messages.

two robots. The first task is a cooperative transportation of a building block from an arbitrary position to a goal. The robots followed planar markers on a obstructed scenario to transport a cube to an ending point also defined by planar markers.



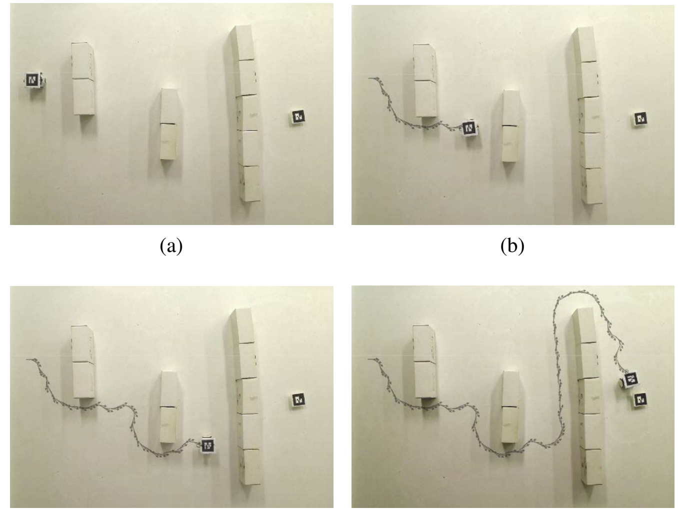
(a) (b) (c) (d)

Fig. 7: Cooperative transportation: (a) Initial configuration with two robots (markers on left and on top) and two goals (markers on middle and at bottom). (b) The first robot moves the cube in the direction of the first marker. (c) The first robot delivers the cube and the second starts to move it towards the next goal. (d) The cube reaches the final goal.

On the second set of experiments (Figure 8) a robot perform autonomous navigation without collisions. A potential field algorithm was used to avoid obstacles and other robots using local sensors (IR proximity sensors). The robot automatically reaches the goal, defined by a planar marker on the robot arena.

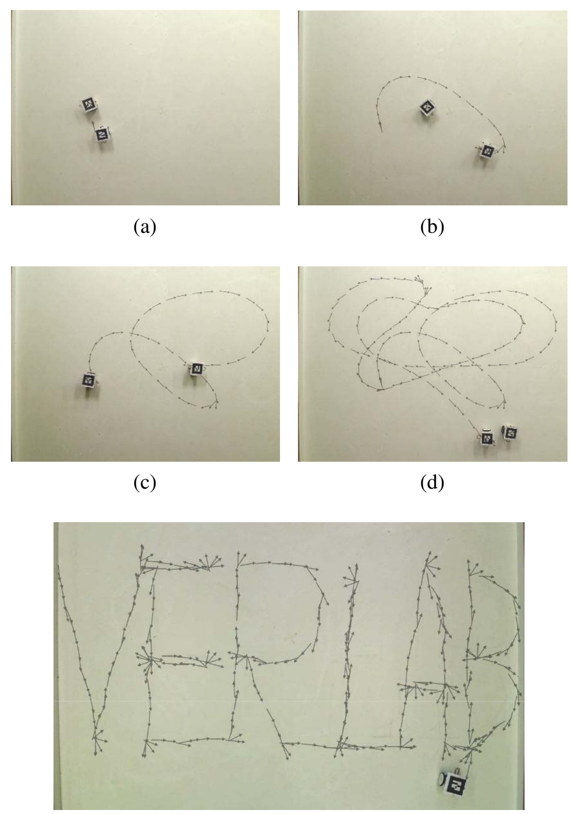
On this third set of experiments (shown in Figure 9) two robots are used to perform a follow-me implementation based on waypoints. This task consists of a tele-operated robot and a robot chaser, which follows the position of the first one. Other waypoint implementation to create a logo with letters can be seen on Figure 9e.

In the last experiment (Figure 10) a robot performs a random walk autonomously without collisions with the



(a) (b) (c) (d)

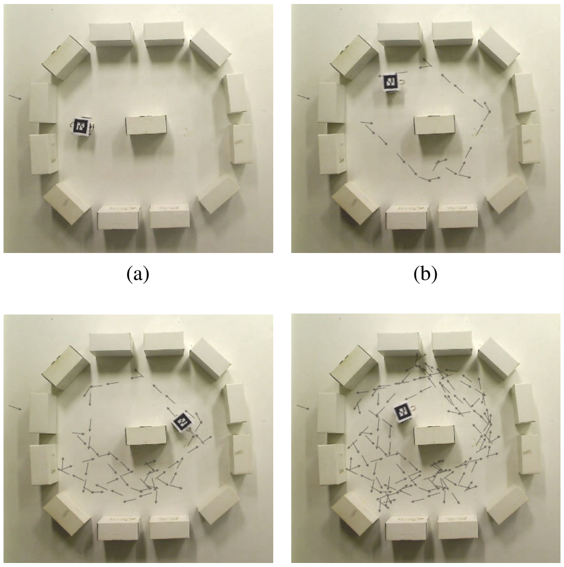
Fig. 8: Potential field: (a) Initial configuration with the robot on left and the goal marker on the right. (b) and (c) The robot navigates around the obstacle avoiding collisions. (d) The robot reaches the goal.



(a) (b) (c) (d) (e)

Fig. 9: Follow me and waypoint: (a) Initial configuration with a tele-operated robot on top and an autonomous robot on bottom. (b), (c) and (d) Once the tele-operated robot moves around the scenario, the autonomous robot tries to follow it. (e) autonomous waypoint following creating a logo.

environment using only on-board information. We placed the robots on several different world configurations to evaluate how well the collision free path and autonomous behaviors were performed.



(a) (b) (c) (d)

Fig. 10: Random Walk: (a) Initial configuration with robot on left. (b) and (c) The robot starts to move around the scenario. (d) After while the robot has explored a large part of the environment.

VI. CONCLUSIONS AND FUTURE WORK

In this work we presented HeRo, a novel small and capable platform for robotic research and education. Specially designed for swarm experimentation, which requires a large set of simpler robots, HeRo is a very low cost alternative to other types of robotic platforms, without removing most capabilities and sensor abilities of commonly used or commercial platforms. The robots and the platform were tested on different scenarios: remote operation, remote control using planar markers and cameras, and autonomous performance. The tests showed that the platform is fast and reliable, and thanks to ROS, it is also easily extendable with the possible development of customized software.

Our future work will focus on improving locomotion accuracy and autonomous capabilities, adding encoders and other types of sensors. We will also improve the 3D design of the robot, making it more robust and easy to modify on-the-fly.

ACKNOWLEDGMENT

This work was developed with the support of UFMG, CNPq, CAPES and FAPEMIG.

REFERENCES

[1] D. Grossman, I. Aranson, and E. B. Jacob, "Emergence of agent swarm migration and vortex formation through inelastic collisions," *New Journal of Physics*, vol. 10, no. 2, p. 023036, 2008.

[2] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, *et al.*, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2, pp. 223–245, 2004.

[3] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal, "Collective transport of complex objects by simple robots: theory and experiments," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 47–54, International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[4] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[5] A. Campo, Á. Gutiérrez, S. Nouyan, C. Pinciroli, V. Longchamp, S. Garnier, and M. Dorigo, "Artificial pheromone for path selection by a foraging swarm of robots," *Biological cybernetics*, vol. 103, no. 5, pp. 339–352, 2010.

[6] G. S. Nitschke, M. C. Schut, and A. Eiben, "Evolving behavioral specialization in robot teams to solve a collective construction task," *Swarm and Evolutionary Computation*, vol. 2, pp. 25–38, 2012.

[7] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 5, pp. 691–701, 2008.

[8] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, 2009.

[9] F. Arvin, J. Murray, C. Zhang, and S. Yue, "Colias: An autonomous micro robot for swarm robotic applications," *International Journal of Advanced Robotic Systems*, vol. 11, no. 7, p. 113, 2014.

[10] M. Le Goc, L. H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, and S. Follmer, "Zooids: Building blocks for swarm user interfaces," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 97–109, ACM, 2016.

[11] F. Arvin, K. Samsudin, and A. R. Ramli, "Development of a miniature robot for swarmrobotic application," *International Journal of Computer and Electrical Engineering*, vol. 1, no. 4, p. 436, 2009.

[12] J. Yu, S. Han, and D. Rus, "A portable, 3d-printing enabled multi-vehicle platform for robotics research and education," *arXiv preprint arXiv:1609.04745*, 2016.

[13] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3293–3298, IEEE, 2012.

[14] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.

[15] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, pp. 59–65, IPCB: Instituto Politécnico de Castelo Branco, 2009.

[16] T. NodeMCU, "Nodemcu-an open-source firmware based on esp8266 wifi-soc," *URL http://nodemcu. com/indexen. html*, 2014.