

The Khepera IV Mobile Robot: Performance Evaluation, Sensory Data, and Software Toolbox

Jorge M. Soares

Iñaki Navarro

Alcherio Martinoli

Abstract—Taking distributed robotic system research from simulation to the real world often requires the use of small robots, which can be deployed and managed in large numbers. This has led to the development of a multitude of these devices, deployed in the thousands by research labs worldwide. This paper looks at the Khepera IV mobile robot, the latest iteration of the Khepera series. This full-featured differential wheeled robot provides a broad set of sensors in a small, extensible body, making it easy to quickly test new algorithms in compact indoor arenas. We describe the robot in detail and conduct an independent performance evaluation, providing test results for all sensors. We also introduce the Khepera IV Toolbox, an open source software framework meant to ease application development. In doing so, we hope to help potential users better assess the suitability of the Khepera IV for their envisioned applications and to reduce the overhead involved in getting started using the robot.

I. INTRODUCTION

A wide range of mobile robotic platforms can be found nowadays in the market and across robotic laboratories worldwide. Some of them have been used by many researchers, reaching a critical mass that elevated them to *de facto* standards in their domain of use [1]–[4]. Convergence to these shared platforms has been argued to improve collaboration and repeatability, allowing for easy validation or refutation of algorithms under study [5], [6].

The success of a robotic platform does not depend solely on its technical qualities, but also on set of accompanying tools, such as libraries, management scripts, and suitable simulators. Among small indoor robots, one platform that achieved widespread acceptance is the Khepera III [7], [8]. Released in 2006, the Khepera III has seen over 600 sales to 150 universities worldwide and has been used in hundreds of publications. In our lab, it has been successfully employed across diverse research topics, including odor sensing [9], [10], navigation and localization [7], rendezvous and formation control [11], flocking [12], and learning [13].

In this paper we present and test the new Khepera IV robot designed by K-Team¹, the successor to the Khepera III. Released in January 2015, the Khepera IV is a differential wheeled mobile robot with a diameter of 14 cm (see Fig. 1). It is equipped with 12 infrared sensors, five ultrasound sensors,

All authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. {jorge.soares, inaki.navarro, alcherio.martinoli}@epfl.ch

J. M. Soares is with the Laboratory of Robotics and Systems in Engineering and Science, Instituto Superior Técnico, University of Lisbon, Av. Rovisco Pais, 1049-001 Lisboa, Portugal.

¹<http://www.k-team.com>



Fig. 1. The Khepera IV robot (image courtesy of K-Team).

two microphones, and a camera. Proprioceptive sensors include two wheel encoders and an inertial measurement unit (IMU). Wireless communication can be accomplished using Bluetooth or 802.11b/g. Processing takes place in an Gumstix embedded computer running GNU/Linux.

We perform an exhaustive test of the sensors and actuators in order to understand their performance and create an accurate model of the robot. The data collected in the process will be made freely available to other researchers, who will be able to use it for deriving and validating their own models. In addition, we present an open source toolbox designed in our lab, composed of a collection of scripts, programs and code modules for the Khepera IV robot, enabling fast application development and making it easier to run multi-robot experiments.

We believe that these performance results, together with the resulting data obtained and the software toolbox are of great interest to potential users of the Khepera IV.

The remainder of this article is organized as follows. In Section II we describe in detail the Khepera IV. Section III focuses on an exhaustive performance test of the different sensors and actuators of the robot. In Section IV we introduce two software packages that complement the Khepera IV. Finally, Section V draws the conclusions about the Khepera IV robot.

II. TECHNICAL DESCRIPTION

The Khepera IV is a small differential wheeled robot designed for indoor use. It is shaped as a cylinder, with a diameter of 14.08 cm and a ground-to-top height of 5.77 cm (including wheels). Its outer shell is composed of two hard plastic parts with slots for the sensors and actuators. Figure 2

shows the technical drawings for the robot. Inside, it follows a stacked PCB design. The complete robot weighs 566 g.

The two actuated wheels are 42 mm in diameter (including the O-rings that act as tires) and are centered on each side of the robot, spaced 10.54 cm apart. Two caster ball transfer units, at the front and at the back, provide the remaining contact points. This solution results in 0.5-1 mm of ground clearance, making the robot very stable but preventing its use on any surface that is not effectively flat and smooth.

A. Electronics

The brain of the robot is a Gumstix Overo FireSTORM COM, an off-the-shelf embedded computer that carries a Texas Instruments DM3730 800MHz ARM Cortex-A8 Processor with a TMS320C64x Fixed Point DSP core, 512 MB of DDR LPDRAM, and 512 MB of NAND Flash memory. The robot ships with a pre-installed 4 GB microSD card for user programs and data. A Wi2Wi W2CBW003C transceiver provides both 802.11b/g (WiFi) and Bluetooth 2.0+EDR capabilities using internal antennas.

Low-level aspects of the robot are managed by a dsPIC33FJ64 GS608 micro-controller that builds a bridge between the embedded computer and the built-in hardware. Additional devices can be connected via an extension bus on top of the robot, as well as an external USB port.

Energy is provided by a 3400 mAh 7.4 V lithium-ion polymer battery. The battery is not swappable and can be charged in approximately 5 hours using the charging jack, but support is also provided for charging from the extension bus (allowing for the use of external, stackable battery packs) and from a set of contacts under the body of the robot (designed for automatic charging stations).

B. Sensors and actuators

The Khepera IV robot is equipped with a rich set of sensing devices:

- Twelve Vishay Telefunken TCRT5000 reflective optical sensors. Eight of these sensors are equally spaced in a ring around the robot body, while four of them are downward-facing. When in proximity mode, the sensors emit a wavelength of 950 nm and their published range is 2-250 mm. They may also operate in passive mode and measure ambient light. The sampling frequency for the infrared sensors is 200 Hz, regardless of the mode of operation.
- Five Prowave 400PT12B 40 kHz ultrasonic transceivers. The sensors' published range is of 25-200 cm with a beam angle of 85° at -6 dBm, and a sensor can be sampled every 20 ms. The effective sampling rate depends on the number of sensors enabled, ranging from 50 Hz for a single sensor to 10 Hz if the whole set is in use.
- A center-mounted single-package ST LSM330DLC iNEMO inertial measurement unit (IMU), featuring a 3D accelerometer and a 3D gyroscope. The accelerometer is configured to a ± 2 g range and a 100 Hz data rate, and the gyroscope is configured to a ± 2000 dps range and a

95 Hz data rate. Data are read by the micro-controller in groups of 10, therefore a new set of accelerometer readings is available every 100 ms and a new set of gyroscope readings is available every 105 ms.

- Two Knowles SPU0414HR5H-SB amplified MEMS microphones, mounted on each side of the robot. The omnidirectional microphones have a gain of 20 dB and a frequency range of 100-10000 Hz. The rated SNR is 59 and the sensitivity is -22 dBV at 1 kHz.
- One front-mounted Aptina MT9V034C12ST color camera with a 1/3" WVGA CMOS sensor, yielding a resolution of 752x480 px. The robot comes with a fixed-focus 2.1 mm lens with IR cut filter, mounted on a M12x0.5 thread. The specified fields of view are 150° diagonal, 131° horizontal and 101° vertical.

Motion capabilities are provided by two Faulhaber 1717 DC motors, one driving each wheel. Each motor has 1.96 W nominal power, transferred through two gearboxes with 38:1 overall gear ratio and 66.3 % overall efficiency, yielding 1.3 W usable power per wheel. The motors are paired with Faulhaber IE2-128 high-resolution encoders, with a full wheel revolution corresponding to 19456 pulses. This yields approximately 147.4 pulses per millimeter of wheel displacement. The motor speed is regulated through pulse width modulation (PWM), and the motors can be set to different modes: closed loop speed control, speed profile control and position control, as well as open loop.

The robots are equipped with three RGB LEDs, mounted on the top of the robot in an isosceles triangle, with light guides to the top shell. The LED color can be controlled with 6-bit resolution on each channel, making them useful for tracking and identification. Finally, a PUI Audio SMS-1308MS-2-R loudspeaker, with nominal power 0.7 W, SPL 88 dBA and frequency range 400-20000 Hz can be used for communication or interaction.

C. Extension boards

The native functionality of the robot can be extended through the use of generic USB or Bluetooth devices, or by designing custom boards plugging into the KB-250 bus. This 100-pin connection provides power, I²C, SPI, serial and USB buses, as well as more specific lines for, e.g., LCD or dsPIC interfacing. K-Team commercializes several boards, including a gripper, a laser range finder, and a Stargazer indoor localization module.

The interface is compatible with that of the Khepera III, and existing boards should work with no alterations. Our lab has, in the past, developed several boards that we are now using with the new robot, including an infrared relative localization board [14], an odor sensing board [15] and a 2D hot-wire anemometer [15], as well as a power conversion board. Figure 3 shows a robot equipped with our custom stack.

Mechanically, however, the different shape of the robot shell may require changes to existing hardware. Boards with large components on the underside can be paired with an additional spacer, while boards inducing significant stress on

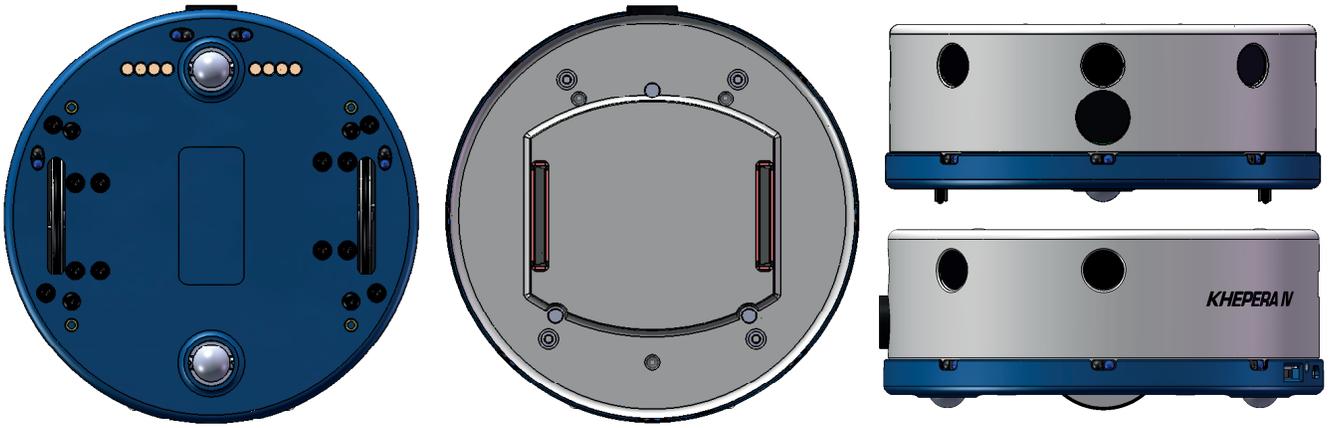


Fig. 2. Bottom, top, front, and left views of the robot (image courtesy of K-Team).



Fig. 3. The Khepera IV robot with custom relative position, odor and wind measurement extension boards.

the connectors should be attached either magnetically or using the screw-in mounting points. Depending on their size and construction, boards may obstruct the view for the tracking LEDs.

III. PERFORMANCE ASSESSMENT

A core part of this paper is the evaluation conducted for the Khepera IV robot and its sensors. This work serves two purposes: informing potential users of the expected behavior and performance of each component, and allowing for the development of robot models.

To this effect, we have undertaken a diligent effort to test all relevant sensors and actuators. We benchmarked the on-board computer, providing an idea of how much algorithmic complexity the robot can handle. We determined the response of the infrared and ultrasound sensors, to determine operational ranges and make it possible to define sensor models. We looked into the camera distortion and the microphone directionality. We assessed the accuracy of the odometry, and provided a superficial analysis of the IMU signals. We have also tested the motor response and the robot's energy use.

While data is presented here in summarized form, we will make the datasets for each experiment available upon

TABLE I

NBENCH RESULTS FOR KHEPERA IV, KHEPERA III AND WORKSTATION

	Khepera IV	Khepera III	Workstation
Numeric sort	212.92	168.76	2218.40
Fourier	856.75	184.46	43428.00
Assignment	3.65	1.10	52.65

publication. It is our hope that these will be of use to other researchers, both for designing their algorithms for the Khepera IV and in implementing more accurate models for simulation.

A. Computation

The computational performance of the embedded Overo Firestorm COM computer was assessed using nbench, a GNU/Linux implementation of the BYTEmark benchmark suite. The same tests were run in the Khepera IV, the Khepera III and a typical mid-range desktop computer, equipped with an Intel Core i7 870 CPU. For both Khepera robots, we used the precompiled binaries available on the Gumstix ipkg repositories, while for our reference computer the program was compiled from source using gcc 4.8.2. The results are presented in Table I.

The three tests are respectively representative of integer arithmetic, floating point arithmetic and memory performance. The benchmark evaluates single-core performance, and therefore does not benefit from the additional cores on the desktop machine. Furthermore, it is not optimized for the DSP extensions on the FireSTORM.

In comparison to the Khepera III, there is a very significant increase in floating point and memory performance, enabling the implementation of more complex algorithms on-board. However, performance is still limited when compared to a desktop computer, which may justify offloading computations, e.g., if undertaking extensive video processing.

B. Infrared sensors

To determine the response of the infrared sensors, the robot was placed in a dimly lit room next to a smooth, light-colored wall, with the front sensor directly facing the wall. The robot

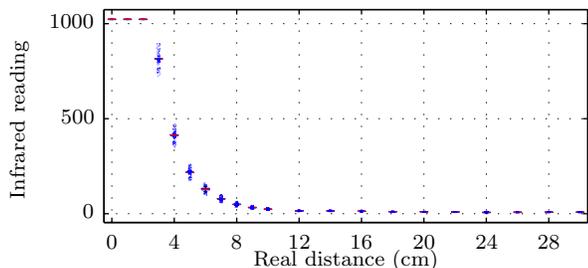


Fig. 4. Box plot of the infrared sensor response.

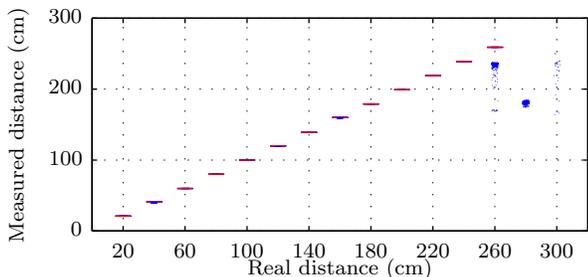


Fig. 5. Box plot of the ultrasound sensor response. For distances greater than 260 cm, the sensor consistently return the code for no-hit (1000) with only outliers present in the plot.

was moved along a perpendicular line to the wall, in steps of 1 cm from 0 cm (i.e. touching the wall) up to 10 cm, and in steps of 2 cm up to a maximum distance of 30 cm. For each distance, 5000 sensor readings were collected. The data is presented in Fig. 4 in box plot form.

Due to the very low variation in readings, most boxes in the plot degenerate to line segments. Below 4 cm, the sensor saturates at the maximum reading (1000). In the range of 4-12 cm the response follows a smooth curve and, for longer distances, the measured value is indistinguishable from background noise in the absence of obstacles.

C. Ultrasound sensors

A similar protocol was followed for the ultrasound sensors, measured against the same target. All sensors were disabled except for the front-facing one, in order to maximize the sampling rate. The robot was placed at distances from 20 cm up to 300 cm, in steps of 20 cm. For each distance, 5000 measurements were obtained. The data is presented in Fig. 5 in box plot form.

The sensor is accurate and precise, with typical sub-cm standard error across the entire published range. Above the 250 cm published range, sensor performance degrades rapidly, with large numbers of ghost detections, generally the product of ground reflections. From 280 cm, the sensor mostly reports no hits, as expected. During the initial experiments, we observed some problems with multiples of the actual distance being returned when the obstacle was positioned at 60 cm distance. These appear to be due to multi-path additive effects involving the floor, and disappeared when tested with



Fig. 6. Example image capture by the robot camera.

TABLE II
CAMERA CALIBRATION PARAMETERS, USING BROWN'S DISTORTION MODEL.

Focal length	F_x	380.046
	F_y	379.116
Principal point	C_x	393.38
	C_y	273.695
Radial distortion	K_1	-0.332931
	K_2	0.113039
	K_3	0
	P_1	-0.000124994
	P_2	0.000195704
Skew		0

a different floor material.

We experimentally determined the ultrasound sensor beam angle, which is of approximately 92° at 1 m, matching the specifications.

D. Camera

An example image capture using the robot camera in a well-lit room is presented in Fig. 6.

3DF Lapyx was used to process a set of 33 full-resolution (752 x 480 px) images of a checkerboard pattern taken with the robot camera and extract its intrinsic calibration parameters. The calibration results using Brown's distortion model are included in Table II. Over the entire set of images, this calibration results in a mean square reprojection error of 0.251978 pixels.

E. Microphones

The microphones were tested using an omni-directional sound source emitting a 1 kHz tone. The robot was placed one meter away, and slowly rotated in place while capturing both microphones. The resulting wave files were bandpass filtered to remove motor noise and extract the reference tone. Figure 7 shows the directionality diagrams for each microphone, with a clear lateral main lobe for each microphone.

F. Odometry

The odometry was tested by having the robot move a significant distance while calculating its position by integrating the

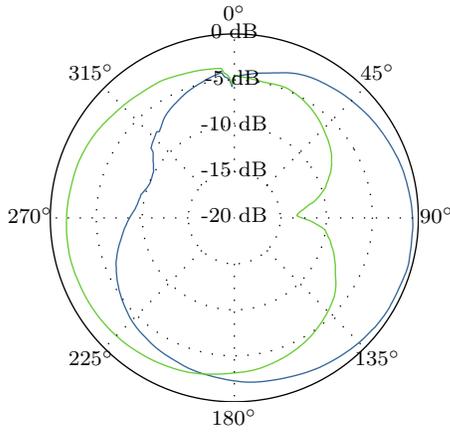


Fig. 7. Directivity pattern for the left and right microphones using a 1 kHz source. The maximum recorded amplitude was taken as the reference.

wheel encoder data. Two paths were tested: a square with one-meter sides, and a circle with one-meter diameter. Multiple experiments were run for each path, with robots moving at approximately 20 % of the top speed. The surface, wheels and rollers were cleaned, and the odometry was calibrated before the experiments. The calibration was performed as described in the odometry calibration example from [16], which consists of a simplified version of the Borenstein method [17].

An overhead camera was used with SwisTrack [18] to capture the ground truth at 20 Hz, while the robot odometry was polled at 100 Hz. The camera was calibrated using Tsai’s method, reporting a mean distance error of 2 mm over the 16 calibration points. A realistic estimate of the maximum error across the entire arena is in the order of 1 cm.

The origin of the trajectory was matched by subtracting the initially measured position from the ground truth tracks, and the initial heading was matched by minimizing the cumulative absolute error over the first 5 % of position measurements. The error metric is the Euclidean distance between the position estimated using the odometry and the actual position of the robot.

The square experiment consisted of describing five laps around a square with side length 1 m, totaling 20 m per experiment (discounting in-place corner turns). The trajectory was programmed as a set of waypoints, with the position being estimated using the odometry alone. As such, in this test, the odometry information is used in the high-level control loop. An example run and resulting absolute error is presented in Fig. 8.

The circle experiment consisted of five minutes spent describing a circle of diameter 1 m, totaling approximately 14.5 laps and 46 m. The robot was set to a constant speed at the beginning of the experiment, and the odometry was logged but not used for the high-level control. The encoder information is, however, still used by the motor PID controller to regulate the speed. An example run and resulting absolute error is presented in Fig. 9.

Each experiment was repeated five times, in both the

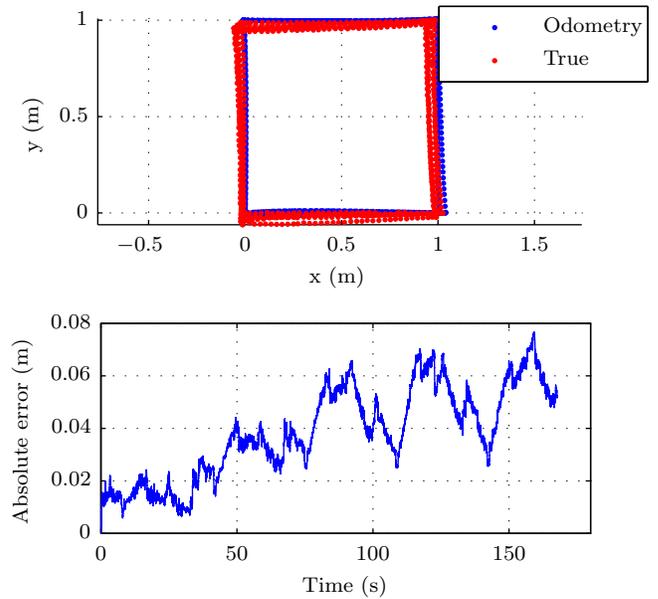


Fig. 8. Odometry-derived and ground truth tracks of the robot while describing four laps around a one-meter square, and associated absolute error over time.

TABLE III
MEAN AND MAXIMUM ABSOLUTE ERROR FOR THE ODOMETRY EXPERIMENTS. EACH ROW IS THE RESULT OF FIVE RUNS.

		Average error		Maximum error	
		μ (m)	σ (m)	μ (m)	σ (m)
Square	Clockwise	0.033	0.006	0.105	0.063
	Counterclockwise	0.033	0.006	0.104	0.038
Circle	Clockwise	0.056	0.031	0.127	0.067
	Counterclockwise	0.093	0.027	0.206	0.054

clockwise and counterclockwise directions. The results are summarized in Table III. For every set of experiments, we take both the average absolute error over the trajectory and the maximum recorded error.

Position estimation using odometry is, by nature, subject to error accumulation over time. However, and while the error does show an upward tendency, it is obvious from figures 8-9 that it is not monotonically increasing. As such, the average and maximum error provide more useful information than the final error.

The error is larger for the circle experiments, as is the error variation. This is partially expected, due to the longer experiment length and the larger fraction of circular motion.

The mechanical set-up of the robots appears to be very sensitive to minor imperfections or dirt on the floor. Namely, the spheres easily get clogged after some hours of use, and even in seemingly flat surfaces the robot sometimes loses traction, significantly reducing odometry performance. Nevertheless, the odometry is very accurate, typically achieving maximum errors in the order of 0.5 % of the distance traveled.

G. IMU

Inertial sensors were tested by having the robot describe a square trajectory similar to the one used in the previous

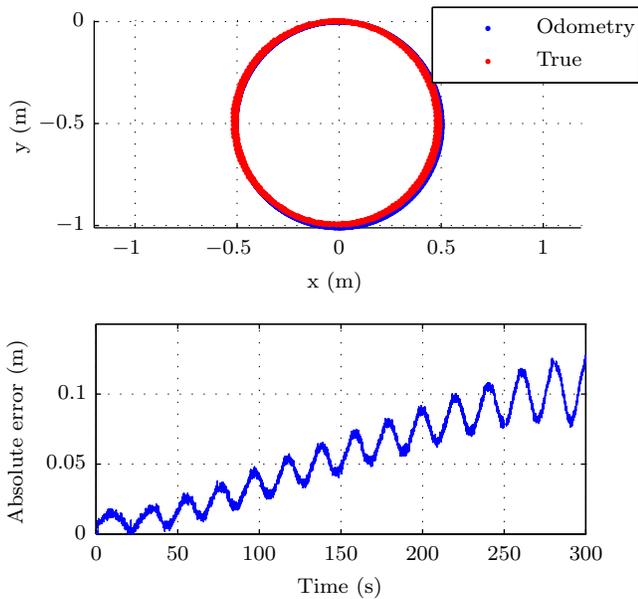


Fig. 9. Odometry-derived and ground truth tracks of the robot while going in a one-meter diameter circle for five minutes (approximately 14.5 laps), and associated absolute error over time.

section, with added pauses before and after each in-place corner turn. The sensors were logged at their maximum frequency, and the captured signals are plotted in Fig. 10. Separately, the scale and bias of the accelerometers were calibrated using the procedure in [19], while for the gyroscopes the initial bias averaged over 200 samples was subtracted from the measurements.

There is no visually apparent structure in the accelerometer data, while in the gyroscope data the corner turns are easily observable. The fact that the robot is inherently unstable in pitch, due to the 4-point support design, creates significant noise in the $z-x$ accelerometer measurements due to the changing orientation of the gravity vector.

Superficial analysis of the data suggests that the IMU can be used for pitch and roll estimation and, while the robot is not equipped with an absolute heading sensor, the yaw delta can be estimated over short time frames. Position and velocity estimation, on the other hand, was not accurate enough to be of use even when using Kalman-based techniques with zero-velocity updates [20].

Given the high quality odometry, the IMU seems of little use for single-sensor dead-reckoning but may complement the odometry in a sensor fusion approach. Perhaps more realistically, it can be useful for attitude estimation, alternative interaction modes, and vibration monitoring.

H. Motors

Figure 11 shows the propulsion power curve, measured for a robot rotating in place at different speed commands. Starting at a baseline consumption of 2.95 W with the robot idle, power increases to a maximum of 5.63 W for speed 800, then decreasing to 4.64 W at full speed.

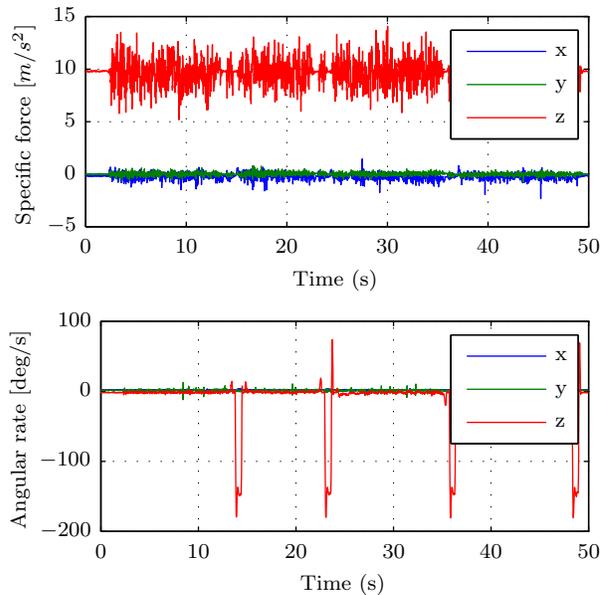


Fig. 10. Accelerometer and gyroscope signals for a single square path. The robot starts moving at $t = 2$ s and briefly pauses at the end of each segment. The peaks in z angular rates correspond to the corners of the square.

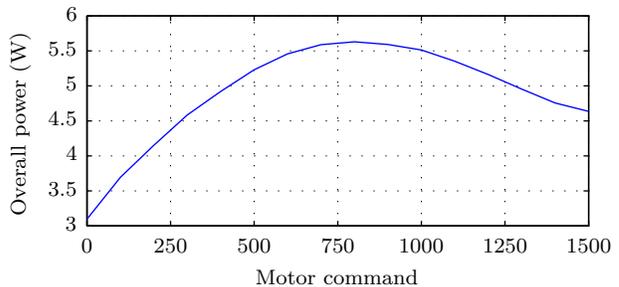


Fig. 11. Self-reported overall power as a function of motor speed commands. Propulsion power is the component over the 2.95 W baseline.

The maximum speed achievable in speed control mode was determined by having the robots move in a straight line while tracking their position with SwisTrack. For a speed command of 1400, the robot achieved a speed of 0.93 m/s. At higher speed commands, approaching the saturation limit of 1500, the robot is unable to move in a straight line and the trajectory begins to curve, with no appreciable increase in linear speed.

The high torque-to-weight ratio allows the Khepera IV to quickly accelerate to the desired speed. Figure 12 shows the results of requesting the maximum speed at time $t = 0.1$ s, obtained by sampling the wheel encoders at 100 Hz period. The robot accelerates to 90 % of the top speed in 0.24 s, and achieves top speed in 0.339 s. This results in an average acceleration $a = 2.74 \text{ m/s}^2$ to 100 %.

I. Power

The number and type of devices activated and in use can significantly influence energy use and limit autonomy. To estimate the potential impact, we used the built-in power

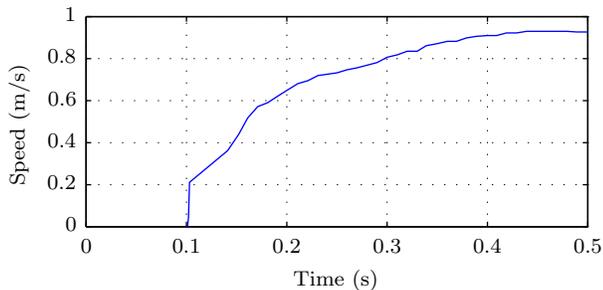


Fig. 12. Motor speed over time, following a maximum speed request at $t = 0.1$ s. The speeds were obtained by capturing the encoder values differences at 100 Hz.

TABLE IV
SELF-REPORTED OVERALL POWER FOR DIFFERENT ACTIVITIES.

Activity	Power (W)
Idle	2.95
CPU load	3.12
Motors (50%)	5.58
Motors (100%)	4.63
Camera	3.12
Ultrasound	3.00
Infrared	2.95
IMU	2.95

management features to measure the energy use when performing different activities. Note that the numbers in Table IV do not necessarily reflect the power used by individual components; for instance, when using the camera, the increase in power is mostly justified by the heavy load placed on the CPU.

Some sensors, such as the infrared and IMU, use negligible energy when actively queried. The largest chunk of power corresponds to idle consumption and is independent of the devices in use. This is, in large part, due to the number of devices that are, by default, initialized on boot, including the 802.11 and Bluetooth radios. Motor power is strongly dependent on the load, and will vary depending on the robot weight, terrain, changes in speed, and obstacles in contact with the robot.

We have also performed a long-term test intended to assess the maximum autonomy of the robots, both in an idle situation (no devices active or programs running) and in a full load situation (camera video streaming to computer, motors moving at full speed, ultrasound sensors active). The results of the test are shown in Fig. 13.

An idle robot starting from a full charge fails after approximately 8.7 h of continuous use, while a robot using maximum power last approximately 5.2 h. This is in accordance with previous results showing high idle consumption.

IV. SOFTWARE

There are, at the moment, two open source libraries for Khepera IV application development: K-Team has developed `libkhepera`, which ships with the robots, and our lab has developed the Khepera IV Toolbox, which we make available as an open source package. Both provide similar base functionality, and are improvements on older libraries for

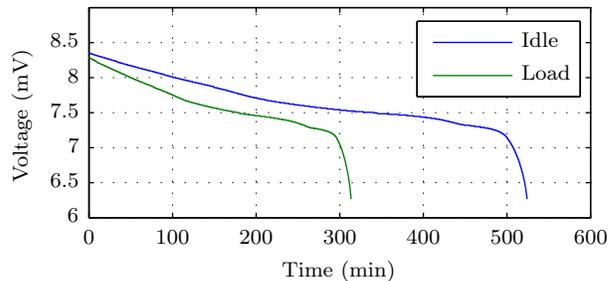


Fig. 13. Battery voltage over time, for idle and full load situations.

the Khepera III robot. The two libraries are independent, and programs using them can coexist in the same robot, although it is not recommended that they run in parallel.

A. `libkhepera`

`libkhepera` is distributed with the Khepera IV. It allows complete control over the robot hardware, generally at a fairly low level. It allows the user to configure devices, read sensor data and send commands. These operations can be accomplished using simple wrapper functions in the main library. Most functions return primitive data types, although some still output data in unstructured buffers.

Conceptually, the library should provide an easy upgrade path for those using the Khepera III equivalent. However, improvements and simplifications in the new library will require changes to existing applications. Full documentation is provided with the library. The outstanding limitation of `libkhepera` is the lack of higher-level constructs, often forcing the user to write verbose code and re-implement frequently used functionality.

B. `Khepera IV Toolbox`

The Khepera IV Toolbox is an evolution of the Khepera III Toolbox [16], with which it shares a significant portion of the code. The initial motivation for its development was providing a straightforward API that could be used with little concern for the underlying details, while also fixing some usability and technical constraints of the robot-provided library.

At a basic level, it provides the same functionality as `libkhepera`, albeit in a different shell. We have developed the API in a way that minimizes the number of lines of code, trying to provide simpler functions that yield, with a single call, the desired result. Most querying functions fill C structures that neatly package complex data.

In addition to the core functionality, the toolbox provides higher level modules that implement support for frequent tasks. These include:

- NMEA, which allows easy processing of NMEA-formatted messages
- Measurement, which handles periodic measurement taking and supports arbitrary data sources
- OdometryTrack, which integrates wheel odometry information to provide a position estimate

- OdometryGoto, which supports waypoint navigation using this position estimate

There are also modules for I²C easy communication, and for each of our extension boards. It is easy to extend the toolbox with additional reusable modules, and the build system makes it trivial to include them in applications.

The toolbox also provides an extensive set of scripts that expedite building, deploying and running applications. These scripts take multiple robot IDs as argument and perform actions such as uploading programs, executing them, and getting the resulting logs, allowing a user to coordinate experiments using relatively large swarms from a single command line.

V. CONCLUSION

In this paper, we have presented the Khepera IV mobile robot and assessed the performance of each of its parts. The robot clearly improves upon its predecessor, packing powerful motors, more complete sensing capabilities and a long lasting battery, all inside a smaller and more stable shell.

The odometry is very accurate in non-slippery floors, making the less precise IMU not very useful for navigation applications. The ultrasound sensors were found to be precise along their entire operating range, while the infrared sensors have somewhat limited range, creating a blind area between 12-20 cm in our experiments; these values depend, of course, on the materials and environmental conditions, and larger ranges can be obtained using specialized infrared-reflective material. The camera and microphones provide good quality information, and are valuable additions to the robot. Among the limitations, the restricted ground clearance has the greatest impact, making the robot unfit for anything but nearly-flat surfaces.

We have also presented two software libraries for the robot, including our own, open source, Khepera IV Toolbox. This library makes it easy to develop applications, and enables the user to easily control multiple robots. It provides a clear upgrade path for users working with the Khepera III robot and corresponding Toolbox.

We will make the Khepera IV Toolbox code publicly available, together with the raw datasets for all our experiments. In this way, we intend to help our colleagues develop their own robot models and jump-start development on the Khepera IV, reducing the platform overhead for future research.

ACKNOWLEDGMENTS

We thank Claude-Alain Nessi and K-Team for their cooperation in the work leading up to this paper and the material provided, and Thomas Lochmatter for his past work on the Khepera III Toolbox and advice.

REFERENCES

- [1] F. Mondada, E. Franzi, and A. Guignard, "The development of Khepera," in *Proceedings of the First International Khepera Workshop*, 1999, pp. 7–14.
- [2] 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*, Portugal, 2009, pp. 59–65.
- [3] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *IEEE International Conference on Robotics and Automation*, 2009, May 2009, pp. 769–774.
- [4] Adept MobileRobots. Pioneer P3-DX. [Online]. Available: <http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx>, [accessed 20-February-2015]
- [5] F. P. Bonsignorio, J. Hallam, A. P. del Pobil, and R. Madhavan, "The role of experiments in robotics research," in *ICRA workshop on the role of experiments in robotics*, 2010.
- [6] L. Takayama, "Toward a science of robotics: Goals and standards for experimental research," in *Robotics: Science and Systems, workshop on good experimental methodology in robotics*, 2009.
- [7] A. Prorok, A. Arfire, A. Bahr, J. Farserotu, and A. Martinoli, "Indoor navigation research with the Khepera III mobile robot: an experimental baseline with a case-study on ultra-wideband positioning," in *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation*, 2010.
- [8] L. Schmidt, B. Buch, B. Burger, S.-h. Chang, J. A. Otto, and U. Seifert, "Khepera III mobile robot practical aspects," University of Cologne, Cologne, Tech. Rep., 2008. [Online]. Available: http://www.uni-koeln.de/phii-fak/muwi/sm/research/k3/download/k3_description.pdf
- [9] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli, "A comparison of casting and spiraling algorithms for odor source localization in laminar flow," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 1138–1143.
- [10] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "A distributed formation-based odor source localization algorithm - design, implementation, and wind tunnel evaluation," in *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA, 2015, [accepted].
- [11] S. Goyal and A. Martinoli, "Real-time optimized rendezvous on non-holonomic resource-constrained robots," in *International Symposium on Experimental Robotics*, ser. Springer Tracts in Advanced Robotics (2013), Vol. 88, pp.353-368, 2013.
- [12] I. Navarro and F. Matía, "A framework for collective movement of mobile robots based on distributed decisions," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 685–697, 2011.
- [13] E. Di Mario and A. Martinoli, "Distributed particle swarm optimization for limited time adaptation with real robots," *Robotica*, vol. 32, no. 02, pp. 193–208, 2014.
- [14] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli, "A fast onboard relative positioning module for multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 151–162, 2009.
- [15] T. Lochmatter, "Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots," PhD Thesis 4628, École Polytechnique Fédérale de Lausanne, 2010.
- [16] ———. Khepera III toolbox (wikibook). [Online]. Available: http://en.wikibooks.org/wiki/Khepera_III_Toolbox, [accessed 20-February-2015]
- [17] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, 1996.
- [18] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "SwisTrack - a flexible open source tracking software for multi-agent systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 4004–4010.
- [19] I. Frosio, F. Pedersini, and N. Alberto Borghese, "Autocalibration of MEMS accelerometers," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 6, pp. 2034–2041, June 2009.
- [20] D. Simn Colomar, J.-O. Nilsson, and P. Händel, "Smoothing for ZUPT-aided INSs," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2012, doi:10.1109/IPIN.2012.6418869.