manuscript No.
(will be inserted by the editor)

Supplementary Materials for the Manuscript Cooperative Object Transport with a Swarm of E-puck Robots: Robustness and Scalability of Evolved Collective Strategies

Swarm Intelligence Journal, Oct. 2016

Muhanad H. Mohammed Alkilabi · Aparajit Narayan · Elio Tuci

Received: date / Accepted: date

Abstract In the experiments described in this paper, homogeneous groups of physical e-puck robots are required to coordinate and synchronise their actions in order to transport a heavy rectangular cuboid object as far as possible from its starting position to an arbitrary direction. The robots are controlled by dynamic neural networks synthesised using evolutionary computation techniques. The best evolved controller demonstrates an effective group transport strategy that is robust to variability in the physical characteristics of the object (i.e., object mass, and size of the longest object's side) and scalable to different group sizes. To run these experiments, we designed, built, and mounted on the robots a new sensor that returns the agents' displacement on a 2D plane. The study shows that the feedback generated by the robots' sensors relative to the object's movement is sufficient to allow the robots to coordinate their efforts and to sustain the transports for an extended period of time. The results of this study demonstrate that robots equipped with a relatively simple sensory apparatus (i.e., no force sensors, and no devices for direct communication) can develop complex group dynamics related to the alignment of pushing forces and the synchronisation of actions during tasks requiring the collective transport of heavy objects. By extensively analysing successful behavioural strategies we illustrate the nature of the operational mechanisms underpinning the coordination and synchronisation of actions during group transport.

Keywords Collective transport · Swarm robotics · Evolutionary computation · Artificial neural networks

M.H.M. Alkilabi, A. Narayan
Aberystwyth University, Aberystwyth, UK
E-mail: {mhm1, apn3}@aber.ac.uk

E. Tuci
Computer Science Department, Middlesex University, London, UK
E-mail: e.tuci@mdx.ac.uk

M.H.M. Alkilabi
Computer Science Department, Kerbala University, Kerbala, Iraq

1 The optic-flow sensors

The optic-flow sensor is an optical camera mounted underneath the robot chassis and located inside the slot originally hosting the robot battery (see Figure 1). We used the commercially available optical mouse sensor ADNS-2610 commonly used to implement a non-mechanical tracking engine for computer mice (Technologies, 2008). The sensor acquires a 18x18 pixels image of the surface with an Image Acquisition System (IAS) made of the HDNS-2100 lens and the HLMP-ED80-XX000 illumination system (see Figure 1). The surface images are captured at 1500 frames per second and sent to the on board Digital Signal Processor (DSP). The DSP computes the pixel displacement in horizontal (x) and vertical (y) direction based on an Optical Navigation Technology. An algorithm accurately computes the optic flow for x and y motion sequences moving at 1 pixel/image. The optic-flow sensor measures a maximum speed of 12 inches per second (ips) with a resolution of 400 counts per inch (cpi) (see Sorensen et al., 2003, for more details). The sensor's readings, after being pre-processed, contribute to form the input vector of the robot controller. The pixels displacement computed in both x and y directions can inform the robot on whether any movements on either the x of the y direction has been achieved in response to the motor commands generated by the controller. Note that in our framework of reference, the y axis is oriented towards to current robot headings (see also Figure 1).

With this optic-flow sensor rotational movements of the robot can not be detected due to the small size of the image. In order to obtain an optimal functionality, the sensor is mounted in such a way so that the distance between the sensor lens reference and the surface is 2.5 mm (see also Figure 2). Since even small variations of this lens/surface distance can cause severe disruptions to the sensors readings, the sensor operability is optimal when the robot acts on flat surfaces. Although the sensor can operates on a large variety of smooth flat surfaces except glasses, it is nevertheless sensitive to different surface colours (see Technologies, 2008, for more details on sensor sensitivity to surface colours).

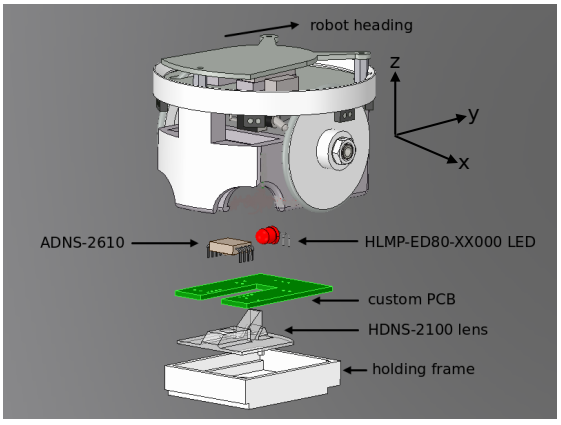


Fig. 1 Drawing of the optic-flow sensor's components and of the e-puck robot. The picture also shows the framework of reference with the sagittal axis (y), the transverse axis (x), and the vertical axis (z).

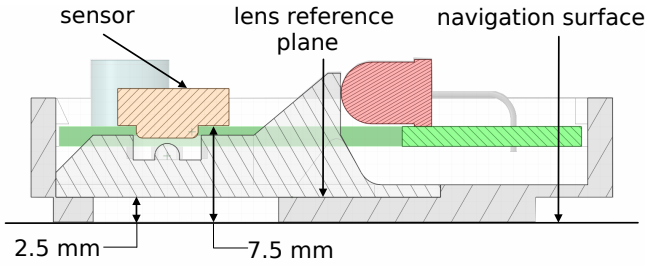


Fig. 2 View of the sagittal section of the robots lower body, showing flow-sensor's components and their distances to the surface.

Figure 3 shows a schema of the hardware components of the optic-flow sensor. The ADNS-2610 communicates the pixels displacement dx and dy to the microcontroller (i.e., Arduino Pro-Mini) via serial communication. The microcontroller, in turn, communicates the dx and dy to the main microcontroller of the e-puck robot through I2C (Inter-Integrated Circuit) communication. The ADNS-2610 sensor uses two 8 bits registers to store the pixels displacements. Reading these registers will clear them. The registers store signed integers in the range $[-128, 127]$. One bit is used to indicate the direction of displacement. In our experiments, the registers are read at the same frequency of the controller updates (i.e., every 130 milliseconds). The optic-flow sensor operates at a higher frequency. The pixels displacements dx and dy are normalised before being passed to the robot controller by dividing them by 128 (i.e., maximum value each registers can hold).

The pixels displacements is communicated to the robot controller in the form of four normalised real values in $[0, 1]$ that in the manuscript are referred to as $+X, -X, +Y, -Y$, with the sign indicating to the direction of displacement on each

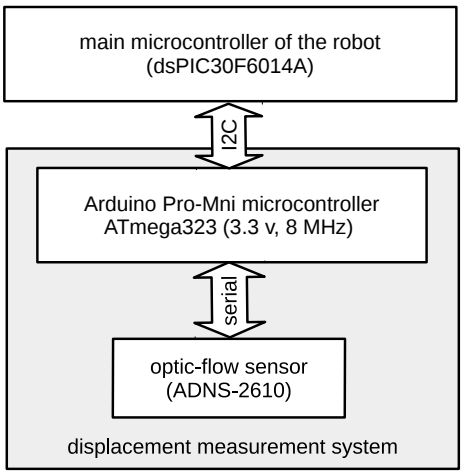
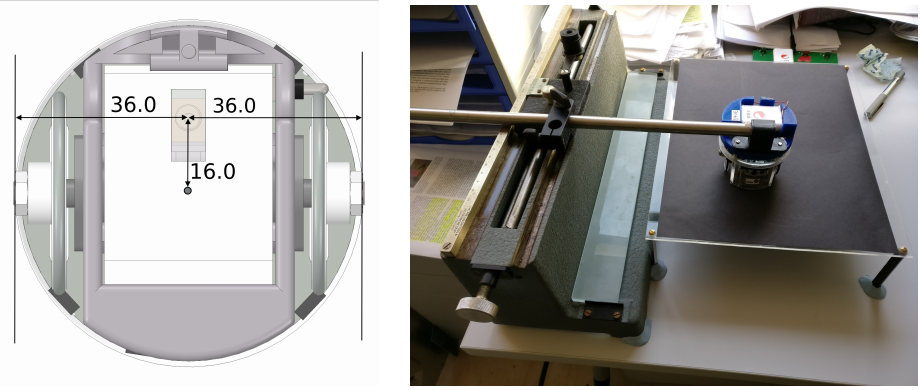


Fig. 3 Diagram showing the main components of the optic-flow sensor and how they communicate with each other and with the e-puck micro-controller.



(a) (b)

Fig. 4 (a) Bottom view of the e-puck robot chassis showing the position of the optic-flow sensor with respect to the robot body. (b) Image showing the robot connected to the microscope linear-stage by an iron bar and a 3D printed connector.

axis. To improve portability of solutions to physical hardware, in simulation, the values of $+X$, $-X$, $+Y$, and $-Y$ are subjected to uniformly distributed random noise in $[-0.025, 0.025]$. The optic-flow sensor generates a sensory stimulus which is a direct feedback on the consequences of the signals sent to the motors. In a collective object transport scenario multiple contingencies can result in a robot failing to execute its desired action. For example, a forward movement command may not produce the desired action if the robot is pushing a stationary object, or an object that is moving in the opposite direction due to forces exerted by other robots. The optic-flow sensor generates readings that can be used by the agents to differentiate between the former and the latter condition and to respond accordingly. The results of our tests show that the extended sensory apparatus of the e-puck robots generates readings that are sufficiently informative to allow the robots to coordinate their effort in order to collectively transport in an arbitrary direction an object that can not be moved by single robots.

1.1 Test to calibrate the sensor's position on the robot chassis

The position of the sensor with respect to the robot body is an important factor that bears upon the accuracy of the sensor readings. In order to minimize the error in the sensor measurements, the sensor has been located as far as possible from the centre of the robot along the y axis, that is 16 mm from the centre of the robot. Figure 4a shows the position of the sensor with respect to e-puck robot body. To verify that the sensor is perfectly aligned with sagittal axis (i.e., y axis) of the robot, we conducted a test in controlled experimental conditions in which the e-puck robot is placed on a black flat surface and connected to a manually operated microscope linear-stage. The robot sagittal axis is aligned with the linear-stage axis ($\theta \simeq 0$) such that any movement of the linear-stage causes the robot to move either forwards or backward (see Figure 4b). During each single trial of this test, the linear-stage is manually moved from the starting position for 250 mm. The direction of movement of the linear-stage is such that in half trials the

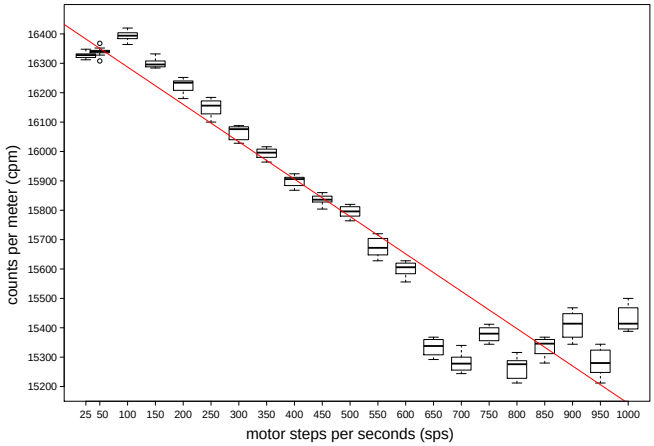


Fig. 5 Graphs indicating the resolution of the optic-flow sensor at different robot speed of motion. For each speed of motion, we have collected 10 measurements of the sensor resolution. The sensor's resolution is expressed in terms of counts per meter (cpm), and the robot speed of motion is expressed in terms of motor steps per second (sps). Boxes represent the inter-quartile range of the data, while the horizontal bars inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. The red line refers to the linear regression model considering only speeds from 25 sps to 1000 sps (equation of the linear regression model $y = -1.272x + 16414.469$).

robot moves forwards and in half trials the robot moves backwards. During each trial, we recorded via Bluetooth the sensor resolution (i.e., the pixel displacements or counts per inch on both the sagittal and traverse robot axis), and the time elapsed to cover the 250 mm. We used the time measurements to verify that in no single trial the robot moved faster than the maximum speed beyond which the optic-flow sensor can not accurately operate (i.e., 12 inches per second). We also verified that no displacements are detected on the transverse axis (i.e., the x axis, the one perpendicular to the linear-stage) at any point during the trial. The results of the test confirmed that the sensor was perfectly aligned with the robot sagittal axis. That is, for any movement of the linear-stage, the optic-flow sensor detected no pixel displacement on the x axis (i.e., $-X = 0$, and $+X = 0$) except for very few circumstances where the sensor detected very small x axis displacements (i.e., $-X \neq 0$, and $+X \neq 0$) caused by the robot shaking while sliding on the surface due to wheel friction. Recall that in this test, the robot is not moved by the rotation of its wheels but it is manually dragged through the linear-stage. The results of this test also suggested that the sensor resolution (i.e., the number of counts per inch, that is, the metric used to refer to the pixel displacement) is sensitive to variability in the motion speed. In other words, we noted that the number of counts is different for different speeds of motion.

1.2 Test to relate sensor readings to actual robot motion

To generate a reliable relationship between the sensor readings and the actual robot displacement, we decided to run a second test whose aim is to generate data to find the relationship between the number of counts of the optic-flow sensor

(i.e., the metric used to indicates the pixels displacements) and the robot linear speed. In each trial of this test, the e-puck robot moves in a straight line on a black surface for a predefined number of motor steps while the optic-flow sensor readings are sampled at 50 milliseconds and sent to a computer connected to the robot via Bluetooth. At the end of each trial, the distance from starting and final position of the robot is measured manually with up to 1 mm accuracy. We conducted a total of 210 trials; that is 10 trials for 21 different robot linear speeds. The graphs in Figure 5 shows the results of this test by plotting the sensor resolution expressed in term of counts per meter (cpm) with respect to the robot linear speed expressed in term of motor steps per second (sps). The graph clearly indicates that the sensor resolution varies with the robot speed.

By running the same test, we have investigated the relationship between sensor resolution and the robot speed of motion for other different optical mouse sensors, namely the ADNS-5020 with a nominal resolution of 1000 cpi, and the A2639B with a nominal resolution of 1200 cpi. Although both sensor have an higher resolution than the ADNS-2610, for both of them a similar relationship between resolution and robot speed of motion has been observed. This phenomena can be attributed to the method used by the sensor to computes displacements. Although the sensor accurately computes the optical flow sequences moving at 1 pixel/image, movements that are greater or less than 1 pixel cause the sensor to fail to correctly determine the actual displacement. Given the variability in the robot speeds such circumstances can occur frequently resulting in variable sensor readings. This results suggests that the optic-flow sensor can be used to compute the distance travelled by a robot within a given interval of time only after having established an accurate relationship between the sensor resolution and the robot speed of motion. In Figure 5, the red line refers to the linear regression model considering only speeds from 25 sps to 1000 sps (equation of the linear regression model $y = -1.272x + 16414.469$).

2 Computational costs

Our experimental design (i.e., number of evolutionary runs, number of genotypes, number of evaluations per genotype, etc.) and the evaluation criteria for physical and simulated robots (i.e., number of trials) are a reasonable compromise between exploration of the problem, and exploitation of the computational resources at our disposal. In the following, we illustrate how we evaluated computational costs and manage the resources to collect the data of this study.

Our simulation models the dynamic aspects of the robot and it's environment (i.e., forces, frictions and torques) using the physics engine "Bullet". The simulation time of an evolutionary process in which solutions are evaluated on a dynamic model of the world can be relatively long. In our study, we have taken all possible measures to limit the simulation time. For example, our code is structured in a way that each evolutionary run makes use of 100 cores of a very large cluster (i.e., the HPC Wales cluster). This means that all 100 solutions of a single generation are evaluated in parallel, using the MPI library. We first estimated, and subsequently verified that to run 20 evolutionary processes, each of which lasting 3000 generations, with 12 evaluation trials for each solution (i.e., 72000000 trials in totals, 12 trials * 100 solutions * 3000 generations * 20 runs) takes slightly less

than the maximum time allotted to a single job (i.e., 72 hours in this cluster). If we add to this the average queuing time of about 4 to 5 days due to cluster usage, and due to the frequent cluster maintenance actions, the elapsed time from job submission to end of all evolutionary runs was on average longer than one week. We considered that any action that could result in a longer simulation time (e.g., more evolutionary runs, more generations, or more evaluation trials) had to be avoided not only for the limit of the 72 hours imposed by the cluster, but also in view of an acceptable experimental design in which hypothesis can be tested and parameters can be set in a reasonable amount of time. On the other hand, we felt that increasing the number of evolutionary runs by shortening the simulation time of a single evolutionary process, like for example by reducing the number of generations per evolutionary run, or the number of evaluations per solution, would have compromised the exploratory process. In conclusion, we believe that our experimental design is a reasonable compromise between exploration of the problem search space, and exploitation of the computational resources.

Time constraints and limits of the computational resources are also the elements we had to take into account in order to set the number of re-evaluation trials. To generate Figure 3b, we run a total of 12800000 trials, given by the following: 2000 solutions evaluated (we evaluated best controller from generation 1000 to 3000) for each of the 20 runs, and 320 trials for every solution (20 trials for every positions * 2 object lengths * 2 object masses * 4 groups cardinality).

For the test whose results are shown in Figure 4 of the manuscript, where we compared the performances of simulated and physical robots, the number of simulation trials is set equal to the number of trials conducted using physical robots. This was done in order to evaluate both systems in identical operational conditions.

3 Graphs of the Fitness Curves of Different Evolutionary Runs

Figure 6 illustrates the fitness curve of 5 different evolutionary runs. Each graph shows the fitness of the best individual (i.e., blue line) and the average fitness of all individuals (i.e., red line) at every generation.

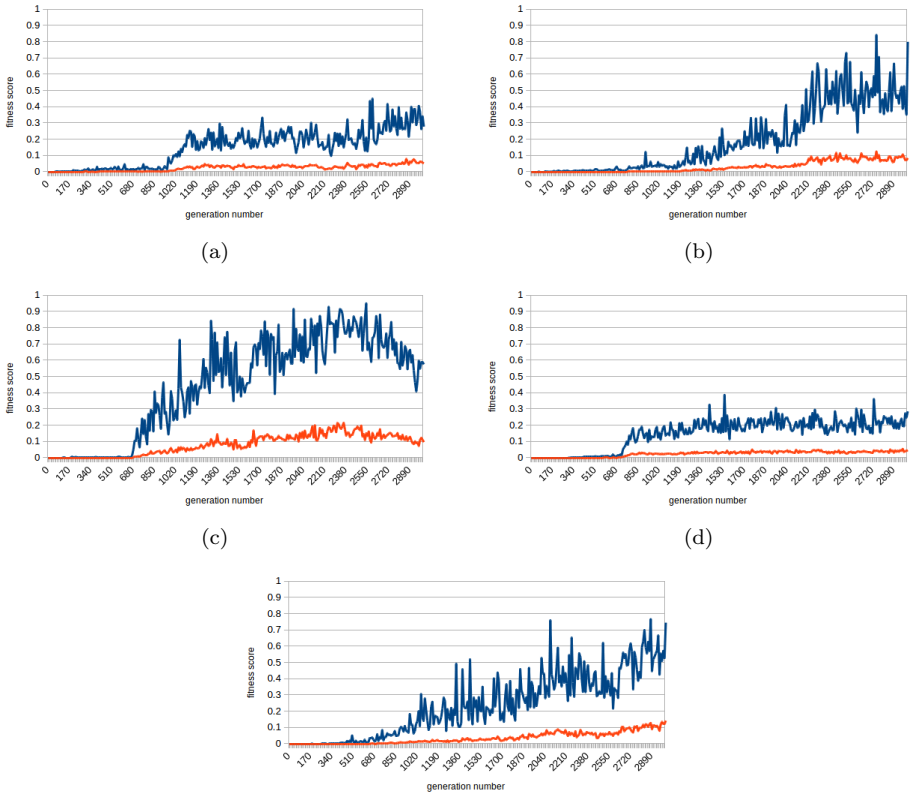


Fig. 6 In each graph, the blue line refers to the fitness of the best group; the red line refers to the average fitness of the population.

4 Movies

4.1 First Movie

The video in Figure 7 shows groups of 3 e-puck robots transporting an elongate cuboid object placed in the centre of a bounded square arena (220 cm side length). The video shows four trials with objects of different physical characteristics:

- trial 1, object of 30 cm length and 450 g mass;
- trial 2, object of 30 cm length and 530 g mass;
- trail 3, object of 40 cm length and 450 g mass;
- trial 4, object of 40 cm length and 530 g mass.

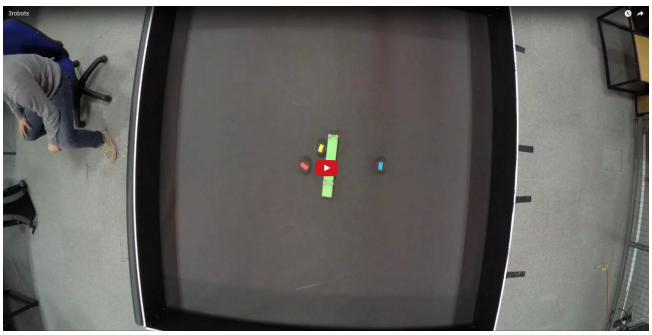


Fig. 7 To play the video, click on the above image or use the following URL `https://www.youtube.com/embed/JMawiG3dDfg`

4.2 Second Movie

The video in Figure 8 shows groups of 4 e-puck robots transporting an elongate cuboid object placed in the centre of a bounded square arena (220 cm side length). The video shows four trials with objects of different physical characteristics:

- trial 1, object of 30 cm length and 600 g mass;
- trial 2, object of 30 cm length and 680 g mass;
- trial 3, object of 40 cm length and 600 g mass;
- trial 4, object of 40 cm length and 680 g mass.

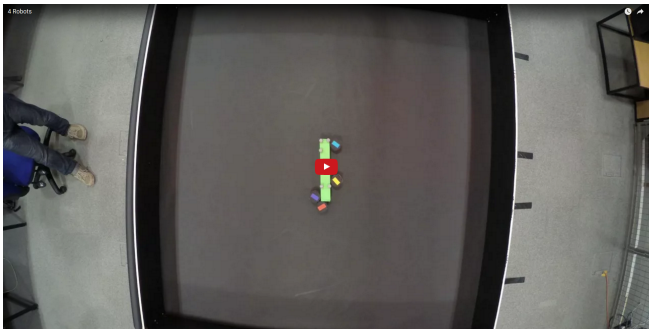


Fig. 8 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/y2qDUpdZajQ`

4.3 Third Movie

The video in Figure 9 shows groups of 5 e-puck robots transporting an elongate cuboid object placed in the centre of a bounded square arena (220 cm side length). The video shows four trials with objects of different physical characteristics:

- trial 1, object of 30 cm length and 750 g mass;
- trial 2, object of 30 cm length and 830 g mass;
- trial 3, object of 40 cm length and 750 g mass;
- trial 4, object of 40 cm length and 830 g mass).

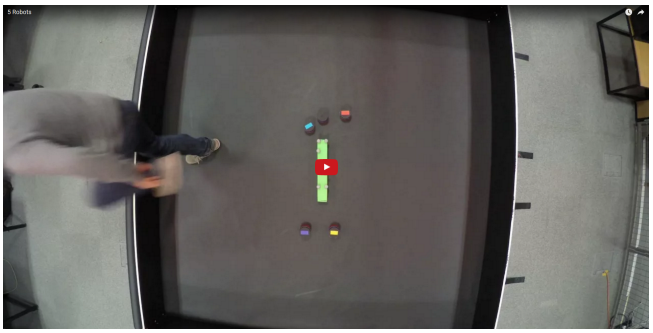


Fig. 9 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/vrXeomAEj0g`

4.4 Fourth Movie

The video in Figure 10 shows groups of 6 e-puck robots transporting an elongate cuboid object placed in the centre of a bounded square arena (220 cm side length). The video shows four trials with objects of different physical characteristics:

- trial 1, object of 30 cm length and 900 g mass;
- trial 2, object of 30 cm length and 980 g mass;
- trial 3, object of 40 cm length and 900 g mass;
- trial 4, object of 40 cm length and 980 g mass.

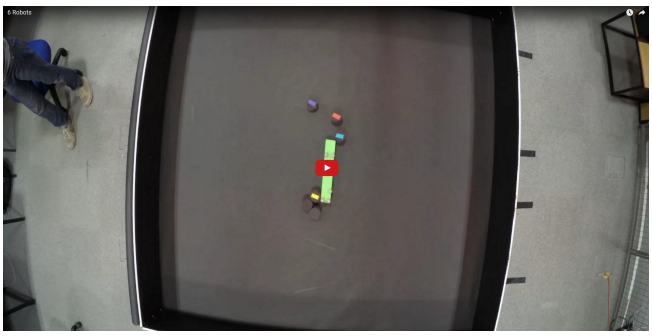


Fig. 10 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/Y-tIwxjWhmU`

4.5 Fifth Movie

The video in Figure 11 shows a robot in action during test A. The video shows two trials with objects of different physical characteristics:

- in trial condition L (light), we use 30 cm length, and 150 g mass object;
- in trial condition H (heavy), we use 30 cm length, and 600 g mass object;

Fig. 11 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/Mf_d3KPpvKo`

4.6 Sixth Movie

The video in Figure 12 shows a robot in action during test B. The video shows two trials with objects of different physical characteristics:

- in trial condition S (short), we use 20 cm length, and 400 g mass object;
- in trial condition R (long), we use 40 cm length, and 400 g mass object.

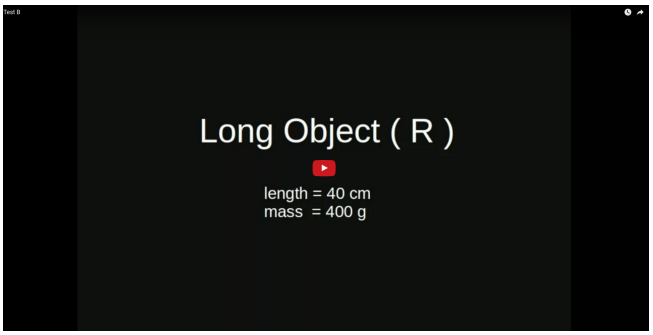


Fig. 12 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/1y0Ci0XOafk`

4.7 Seventh Movie

The video in Figure 13 shows a robot in action during test C. The video shows one trial. The object's length is fixed to 30 cm. The object mass is altered manually by adding/removing an iron bar of 450 g to the object every 60 s interval. In this test, the object mass is heavy during the first 60 s ("H", 600 g), then light for the following 60 s ("L", 150 g), and then heavy again in the last 60 s interval ("H", 600 g).

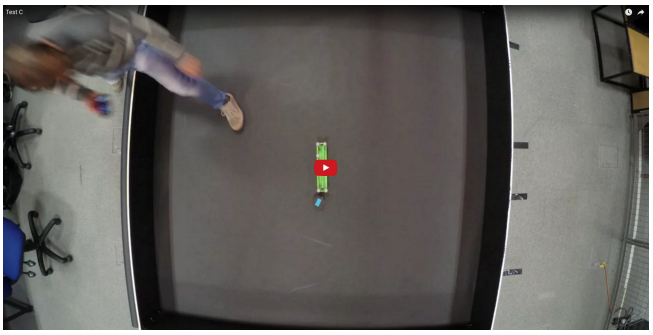


Fig. 13 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/b0QSI3X1h6s`

4.8 Eighth Movie

The video in Figure 14 shows a robot in action during test D. The video shows one trial. The object's length is fixed to 30 cm. The object mass is altered manually by adding/removing an iron bar of 450 g to the object every 60 s interval. In this test, the object mass is light during the first 60 s ("L", 150 g), then heavy for the following 60 s ("H", 600 g), and then light again in the last 60 s interval ("L", 1500 g).

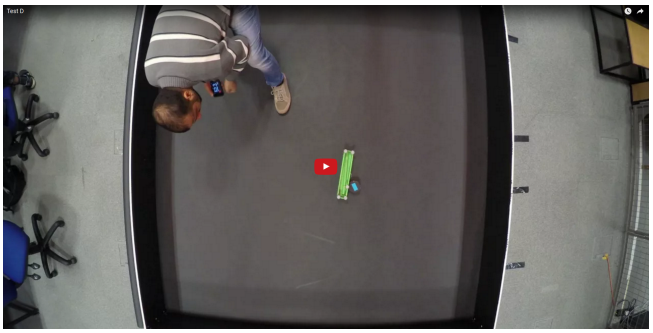


Fig. 14 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/FmS8OXYU72g`

4.9 Nineth Movie

The video in Figure 15 shows a robot in action during a test in which an e-puck robot is placed on one side of the object and two e-puck robots placed on the opposite side of the object. We record the reading from the optic-flow sensor and number of repositioning events from the two front infra-red sensors. In each trial, we counted the repositioning events with pushing forces exerted on the first touched long side of the object. We stopped counting as soon as the robot touches the other long side of the cuboid object.

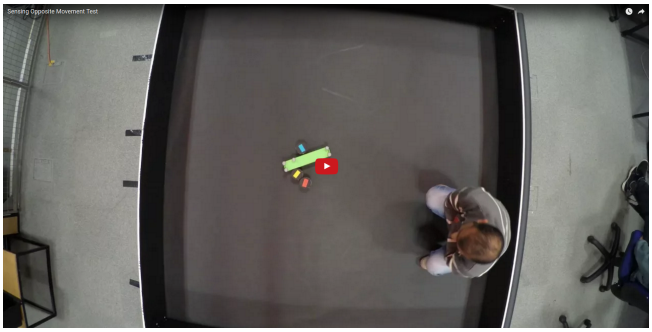


Fig. 15 Video showing sensing the opposite object's movement test. To play the video, click on the image or use the following URL `https://www.youtube.com/embed/h9-y9saF5uo`

4.10 Tenth Movie

The video in Figure 16 shows a group of 16 simulated robots transporting an object of 70 cm length and 2400 g mass.

Fig. 16 To play the video, click on the image or use the following URL `https://www.youtube.com/embed/dR4YOSkYjWE`

5 Sinuosity

Table 1 Table describing the mean and standard deviation of sinuosity of object trajectories, and of duration of transport when the object is pushed by real e-pucks groups controlled by the best evolved controller, in 16 different evaluation conditions. Sinuosity is measured only in successful trials.

group	object		sinuosity		duration (s)		N. Succ.
size	length (cm)	mass (g)	mean	sd	mean	sd	trials
3	30	450	1.87	0.81	71.40	24.05	10
		530	2.74	1.71	70.50	17.86	10
	40	450	1.45	0.38	45.10	13.74	10
		530	3.31	2.17	56.80	13.02	10
4	30	600	1.93	0.35	77.20	29.04	10
		680	2.88	1.12	79.20	26.39	10
	40	600	1.29	0.35	60.90	27.21	10
		680	2.63	0.99	58.00	14.52	10
5	30	750	1.78	0.87	73.20	35.41	10
		830	3.69	3.06	67.20	45.85	8
	40	750	1.61	0.91	44.20	11.30	10
		830	3.55	1.74	95.70	36.99	10
6	30	900	3.11	1.97	109.00	39.31	9
		980	2.38	0.03	95.00	21.21	2
	40	900	2.08	1.02	66.80	32.35	10
		980	4.01	1.68	119.60	36.17	10

References

D.K Sorensen, V. Smukala, M. Ovinis, and S. Lee. On-line optical flow feedback for mobile robot localization/navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1246–1251, 2003.

AVAGO Technologies. Adns-2610 optical mouse sensor, date sheet, 2008. URL `https://media.digikey.com/pdf/Data%20Sheets/Avago%20PDFs/ADNS-2610.pdf`.