Mechatronics xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Encoder-free odometric system for autonomous microrobots

Serge Kernbach

Institute of Parallel and Distributed High-Performance Systems, University of Stuttgart, Universitaetsstr. 38, 70569 Stuttgart, Germany

ARTICLE INFO

Article history:
Received 1 May 2011
Accepted 7 May 2012
Available online xxxx

Keywords:
Motion feedback
Odometry
Autonomous robotics
Self-calibration
Robot swarm

ABSTRACT

This paper describes an odometric system for autonomous microrobots based on the optical properties of a motor–wheel transmission for motion feedback. We demonstrate that infra-red light, reflected from this mechanical system, shows distinctive periodic changes of amplitude, correlated with the motion of the robot. Real-time odometric signal processing, performed on a small embedded microcontroller, allows accurate movement detection for each wheel and is used for motion control, self-calibration, and calculation of differential and absolute velocity, displacement and rotation. We demonstrate some applications of this on-board odometric system in swarm robot experiments. This approach can also be applied to other industrial or academic mechatronic systems, where size or computational power are limiting factors.

1. Introduction

The miniaturization of devices, and especially robots, represents an important trend in current science and technology [12]. The current state of the art in miniaturization of fully autonomous microrobots approximates to robots measuring $2 \times 2 \times 2\,\text{mm}^3$, where the microrobot is capable of local directional communication, perception and precise actuation in the micro-world [9,32]. However, the miniaturization of complex mechatronic systems is challenging in terms of power supply, active perception and actuation, and the achievement of extended functionality.

The functionality of small autonomous devices is extremely limited, primarily by their small size, tiny power storage capacity, and technological constraints. To achieve complex functionality, the microrobotic community exploits a collective approach, in which a large number of small and simple cooperating robots achieve a similar complexity in terms of their collective functionality, adaptivity or computational power [19]. This methodological approach is known as swarm robotics [29] and addresses specific research objectives, such as the spatial and temporal coordination of a large number of microrobots [28], cooperative actuation [33], investigation of bio-inspired [4], and evolutionary strategies [27], modeling [24], robustness [3], self-assembling approaches [13], and other topics. To progress towards these research objectives, we created the flexible, cost-effective and modular "Jasmine" microrobotic platform,[1] see Fig. 1a. More than 300 of these microrobots have been manufactured and distributed between different institutions, allowing large-scale swarm robotic experiments [17].

The spatial coordination of a large-scale system requires the capability for relative position identification and normalized differential and absolute velocities. This functionality is necessary for, for example, trophallaxis [30], foraging [16], and behavioral or other bio-inspired spatial approaches. Developing an odometric system for a microrobot that combines acceptable accuracy, low cost and small size represents a serious challenge. One of the main problems lies in obtaining motion feedback from the wheels [6] or the ground [5]. Using typical transitive/reflective optical, magnetic or capacitive encoders installed on motors or on the wheels is almost impossible, because of size limits.

In this work, instead of using special encoders, we propose the use of certain properties of the motor–wheel transmission: gear-teeth, eccentric rotation of gears and wheels, and electric processes in the pulse-width modulation (PWM) control of motors. We demonstrate that infra-red (IR) light, reflected from the motor–wheel transmission, shows distinctive periodic changes of signal amplitude, correlated with the robot's movement. Information about this movement can be extracted by a real-time signal processor running on-board. However, the small microcontroller, the limited memory (1 kB RAM) and considerable electronic noise make real-time signal processing very problematic. For example, it is impossible to store several periods of the whole signal to apply recursive signal analysis approaches [26] or a Kalman filter [21]. In this work, we focus on Constraint Satisfaction Problem (CSP)-based algorithms for performing real-time odometric signal processing to allow accurate detection of each wheel rotation.

Section 2 of this paper discusses the optical properties of motor–wheel transmission for odometric purposes and considers some solutions. Section 3 deals with real-time odometric signal processing using only 354 bytes RAM and about 1.2 kB flash

E-mail address: Serge.Kernbach@ipvs.uni-stuttgart.de

[1] See www.swarmrobot.org.

0957-4158/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

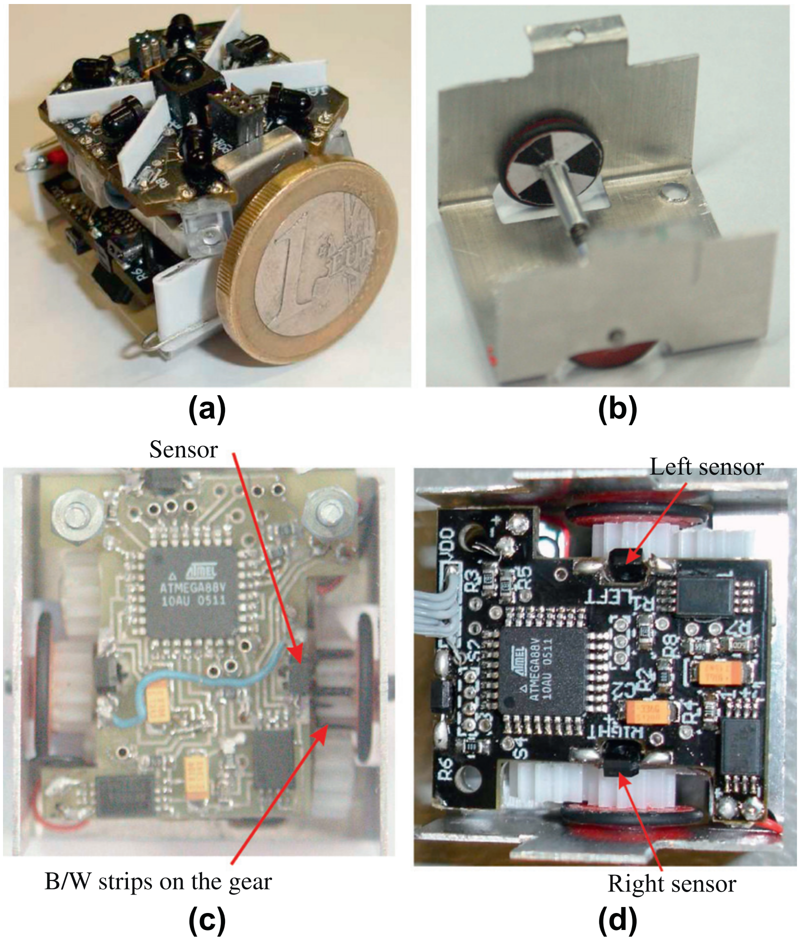


Fig. 1. (a) The "Jasmine" microrobot. (b) The reflective encoder installed in the first version of the system. (c) Geared motor–wheel coupling with painted black and white stripes and optical sensors TCNC1000 installed perpendicularly on the motor PCB. (c) Final placement of sensors (with 60° declination) and gears without stripes.

ROM. In this section, we also describe adaptive plausibility filters and the detection of blocked wheels. Section 4 discusses the calibration strategy for microrobots. Finally, in Sections 5 and 6 we demonstrate our experiments and conclude this work.

2. Optical motion-feedback system in the microrobot

Typically, microrobots of <1 cm^3 do not possess an on-board motion feedback system, for various reasons. Primarily, using piezo-actuators of a precision between a few μm and a few nm does not offer on-board odometric systems of a resolution high enough for any practical purposes. These robots use external vision-based tracking systems for navigation and actuation. Microrobots of >1 cm^3, with DC motors, have different options for on-board odometry: using Hall sensors in motors, analyzing the power consumption of PWM control, vision-based ground tracking systems, or optical/magnetic/capacitive encoders on the wheels. Depending on the limits of the size and power budget of the microrobot, the eventual choice is constrained by optical or magnetic approaches, which do not require large components.

The "Jasmine" microrobot [11,18] shown in Fig. 1a has two small DC motors with integrated planetary gearboxes. The mechanics[2] consist of two plastic gears glued to the motors' and wheels' axes. In this way, the force from the motors is transmitted to each of the wheels. The microrobot[3] has two electronic boards, the motor board and the main board, which communicate through a 200 kHz I2C interface. The main board holds the ATmega 168 microcontroller with 16 kB ROM, 1 kB RAM and is used for six-channel communication, proximity sensing, remote control, geometry and light sensing, energy management, ZigBee communication, and behavioral control of the robot. This board is also used to support the upper extension boards of the robot. The motor board, which holds an ATmega 88 microcontroller, is used for motor control, the odometric system, energy control, and touch and color sensing. It also provides four channels for further sensors/actuators. The robot uses an advanced power management system, with a Li-Po accumulator, which provides for 1.5 h of autonomous work and the capability for autonomous docking to the power station and auto-recharging [16,14].

Since optical or magnetic on-board odometrical systems require corresponding encoders to receive motion feedback from the wheels, we tested several approaches [7,10]. First, we used IR-reflective sensors, which are widely used in robotics, not only for reflective odometric encoders [1] but also for navigation [8], distance measurement [2], and self-assembly tasks [23] in modular systems [22]. Such sensors are also used in micro- [31] and mini- [25] robotic platforms. Second, as well as reflective wheel encoders, see Fig. 1b, we also tested solutions based on analyzing the current consumption of the DC motors, using Hall sensors installed

[2] Developed by Marc Szymanski and Ramon Estana from the University of Karlsruhe.

[3] The first version was 26 × 26 × 20 mm^3, the final version is 30 × 30 × 20 mm^3 with 2 ATmega 328P pico-power microcontrollers.

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

on the motors, and using black and white stripes painted on the gear, see Fig. 1c. After several tests, it became clear that the geared coupling is the best solution, allowing not only effective force transmission from motors to wheels, but also being well suited for the odometric system even without painting the gear, see Fig. 1d.

The idea behind this approach is that by illuminating gears with an IR beam of a small opening angle, the values of the reflected light will depend on the distance between the gear surface and the reflective sensor. In turn, this distance depends on the position of the gear tooth and finally on the rotation of the wheel. Sampling frequently enough (in accordance with the Nyquist theorem), we can reconstruct the signal, and the motion parameters, such as the speed or angular rotation of the microrobot.

The signal received from the IR sensor has two different components, readily visible in Fig. 2. The high-frequency signal depends on the position of the gear tooth. The second, low-frequency, signal is caused by eccentric rotation of the wheel. Even small deviations (less than 0.1 mm) are readily detected by the IR sensor. The period of this signal is exactly one turn of the wheel. In experiments, we encountered high noise levels, produced by the microcontrollers switching communication IR-emitters (each gate switches 20 mA current) and by the H-bridges of both motors, which afforded considerable problems in encoding a high-frequency signal.

Odometric signal processing is based on detecting maxima or minima of signal amplitudes, shown in Fig. 2. The real displacement of the robot, caused by the rotation of the wheel, can be calculated as the relationship of the wheel perimeter to the number N of detected maxima in the signal during the time t. Such detected maxima, or odometrical periods P, are related to the number of teeth in the gear ($N = 12$) in the high-resolution mode and to the size of the whole wheel in the low-resolution mode ($N = 1$)

$$S_h = \frac{\pi D}{N} = \frac{11.5\pi}{12} = 3.0107 \text{ mm}, \tag{1}$$

$$S_l = \frac{\pi D}{N} = \frac{11.5\pi}{1} = 36.1283 \text{ mm}, \tag{2}$$

where D is the diameter of the wheel, and $S_{l,h}$ the corresponding distances related to the period P. By measuring, and correspondingly calibrating, more points n for P, we can achieve a higher resolution. For instance, at a resolution of 0.3 mm in the high-resolution mode and 3.6 mm in the low-resolution mode, it is enough to have $n = 10$ points between two signal peaks. This makes the signal frequency $F_{hn,ln} = nN\frac{v}{\pi D}$, where v is the velocity. Generally, a sampling frequency F_h and F_l can be estimated as:

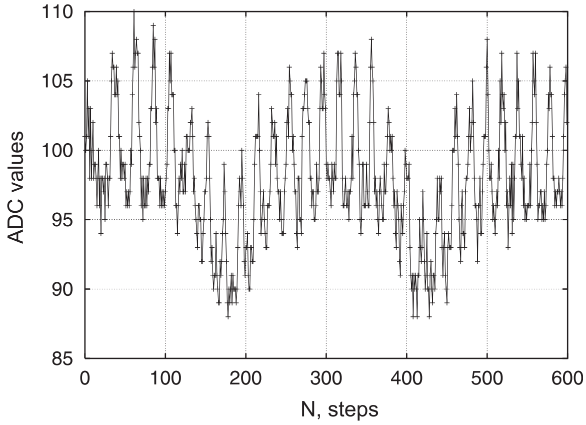


Fig. 2. The signal (without noise), obtained after ADC conversion (denoted as "ADC values" in this and other figures) of the voltage from the reflective IR sensor. The high-frequency component obtained from the gear tooth and the low frequency component caused by the eccentric motion of a gear are visible.

$$F_h \geqslant 2F_{hn} = 2nN\frac{v}{\pi D} = 240\frac{100}{36,1283} = 664.2992 \text{ Hz}, \tag{3}$$

$$F_l \geqslant 2F_{ln} = 2nN\frac{v}{\pi D} = 20\frac{100}{36,1283} = 55.3583 \text{ Hz}. \tag{4}$$

This sampling frequency means that the microcontroller should sample every ~1.5 ms and ~18 ms respectively. The ADC of the ATmega88 microcontroller used (8 MHz clock) with a 250 kHz internal clock requires 13 cycles (52 μs) for a single conversion.

The odometry is integrated into the motor control system, see Fig. 3. The PWM-signals for a defined motion velocity are continuously provided to both H-bridges, which are periodically switched on and off by the timer signals. This allows a saving of about 1/2–2/3 of the energy required for continuously-powered motors. The first time the timer interrupts, it initiates motion for the time *timerMotion*. After *timerMotion* expires, the timer interrupts a second time and stops the motion for the period *timerMeasurement*. During this period the IR-emitters turn on and illuminate the gears. After *timerMeasurement* expires, the timer interrupts again, starts the ADC, turns off the IR-emitters, increases the variable *goneDistance* for both wheels and initiates the motion again. The ADC works in parallel: when the conversion is complete, the corresponding interruption updates the current number of samplings n between two maxima in the signal. In this way, the sampling frequency (F_{ls}) for 250 kHz timer frequency can be calculated (we ignore the constant time of the interruption handler, which is about 5–10 μs) for each wheel separately.

$$F_{ls} = \frac{250,000}{2 * (timerMotion + timerMeasurement)}. \tag{5}$$

As the approach explained in Fig. 3 is applied to each wheel separately, we need to multiply the denominator of Eq. (5) by two. We use *timerMotion* = 70 and *timerMeasurement* = 35, so that the sampling frequency for each wheel is about 1190 Hz (or 0.84 ms). This corresponds to the required high-resolution sampling frequency F_h, and is much higher than the required F_l. This relatively high sampling frequency provides a theoretical resolution of the odometrical system of about 0.18–0.45 mm. Since sampling occurs with the same frequency, only one parameter can change the number of samples n between two maxima of the signal: changes in the velocity of the robot, calculated as

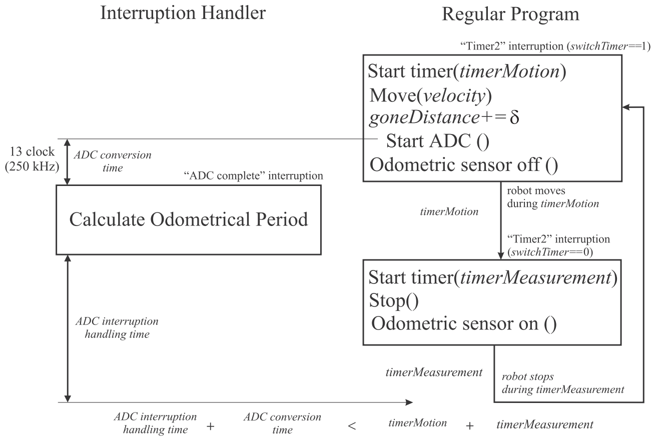

Interruption Handler
Regular Program
"Timer2" interruption (switchTimer==1)
Start timer(timerMotion)
Move(velocity)
goneDistance+=δ
Start ADC ()
Odometric sensor off ()
13 clock (250 kHz)
ADC conversion time
"ADC complete" interruption
Calculate Odometrical Period
timerMotion
robot moves during timerMotion
"Timer2" interruption (switchTimer==0)
Start timer(timerMeasurement)
Stop()
Odometric sensor on ()
ADC interruption handling time
timerMeasurement
robot stops during timerMeasurement
ADC interruption handling time + ADC conversion time < timerMotion + timerMeasurement

Fig. 3. Structure of the algorithm for the odometric system. The function *StartADC()* starts the ADC conversion and the corresponding interruption handler. The variable *velocity* defines the PWM signal on the motors (for example the velocity of the robot), *timerMotion* defines the running time of motors and *timerMeasurement* regulates the exposed time of the IR LEDs (that is, the time the LEDs are on) and thus regulates the intensity of the reflected IR signal. Critical values for the timing of different components are demonstrated below the structure. The value of δ is calculated based on Eq. (7).

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

$$v = \frac{\pi D F_{ls}}{n}. \qquad (6)$$

We observe the number of samples n between 70 and 170 per one rotation of the wheel, which points to a velocity of the robot between 614 mm/s and 252 mm/s. The values

$$\delta = \frac{S_l}{n}, \qquad (7)$$

$$goneDistance = \sum_i \delta_i \qquad (8)$$

represent the current displacement of the robot and the accumulated distance traveled by the corresponding wheel. By controlling *goneDistance* in each wheel, we can determine not only the distance traveled, but also the rotation of the robot.

3. Odometrical signal processing

The odometric signal processing should allow stabilization of motion (the same rotation frequency of both wheels), normalization of velocity (all robots move with the same velocity), normalization of the rotation angle and finally, calculation of the distance traveled by a robot. These four operations can be performed by detecting one period P of the signal (corresponding to one turn of the wheel) in terms of the number n of samples within P. The value n allows the calculation of other parameters defined by Eqs. (6)–(8). For the signal processing, we have only around 600 bytes of RAM and so all calculation should be finished within 600 µs in the interruption procedure on the 8 MHz 8-bit microcontroller.

Fig. 4 outlines the notion of real-time signal processing: We need to detect the *Max* and *Min* peaks in the signal. When a candidate for *Max* is found, this value is temporarily stored. As long as the *Min* value is found, it means that the last *Max* candidate is reliable and can be used for calculation of the next *period P* value. After the next *Max* is found, the last *Min* candidate can be used for calculation of P and the whole cycle repeated.

The algorithm of period detection takes the form shown in Fig. 5. The first task is to detect the plausibility regions for *Max* and *Min*. This is done by storing absolute *aMax* and *aMin* values within the first 250 values obtained from the ADC and calculating $\Delta = (Max - Min)/5$. After that, the values between *Max* and $Max - \Delta$ are set to the upper plausibility region, whereas the values between *Min* and $Min + \Delta$ are set to the lower plausibility region. The values of current *Max* and *Min* will be searched within these intervals. The values for the current *Max*, *Min* are sequentially recognized and the corresponding period P and Δ are calculated. The period will be also tested by an adaptive plausibility filter, which neglects implausible values for P.

It is much easier to detect the lower peaks (*Min*), see Fig. 6; these make the recognition more stable for noise (the upper peaks are most suitable for high-frequency amplitudes). To detect the period, we use a CSP-based approach, where the points ph_i (candidates on *Max*) pl_j (candidates on *Min*) should satisfy the constraints, and are collected in Table 1. The CSP-solver delivers good recognition of peaks (one wrong value from approximately 150 correct values) when the wheel is not displaced mechanically. For this reason we need an adaptive plausibility filter, which checks the values obtained for plausibility.

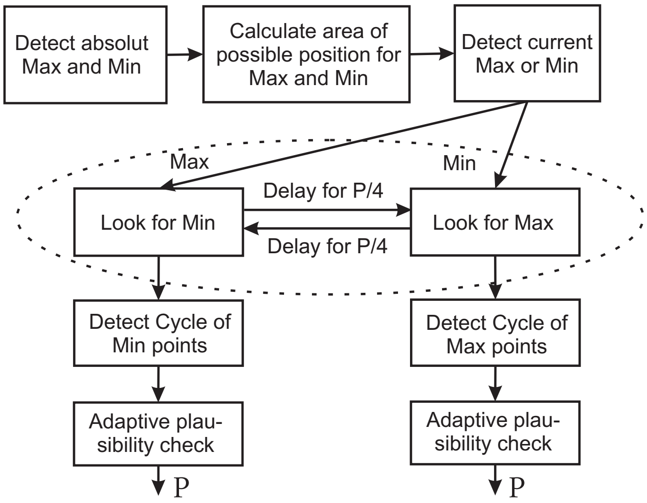

Detect absolut Max and Min
Calculate area of possible position for Max and Min
Detect current Max or Min
Max
Min
Look for Min
Delay for P/4
Look for Max
Delay for P/4
Detect Cycle of Min points
Detect Cycle of Max points
Adaptive plau-sibility check
Adaptive plau-sibility check
P
P

Fig. 5. Algorithm for the period detector.

3.1. Adaptive plausibility filter

Despite good period recognition, distortions (primarily of mechanical origin from sideways wheel displacement) can cause an specific variation in the amplitude of the peaks. In this case, the CSP detector is unable to distinguish the *Min* or *Max* peaks within the noisy signal. This results either in one lost peak (a detected period of double size) or in the appearance of smaller periods. However, this does not always mean failures in the mechanical system; the period can suddenly become smaller or larger, for example when a robot begins moving an object in front of itself. See the experiments in Section 5. Therefore, we need a filter that is able to distinguish "wrong" smaller or larger periods from "correct" ones. The structure of the adaptive plausibility filter is shown in Fig. 7.

The major idea of the filter lies in the observation that incorrect values of P are of a short-term nature, where correct P exist for longer periods of time. The filter tests the values obtained for P

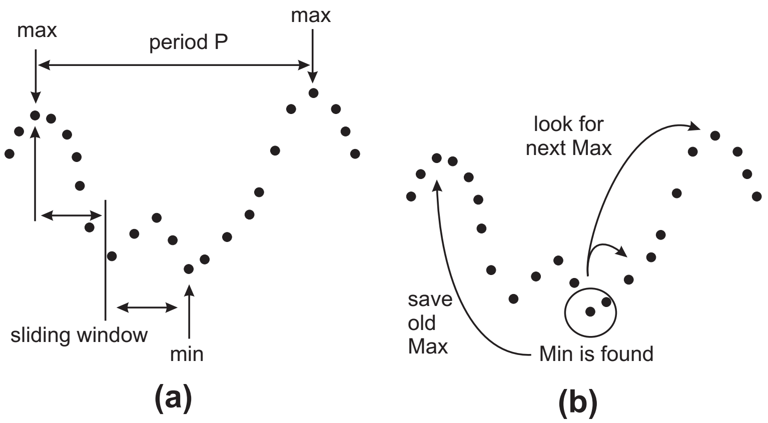


Fig. 4. (a) Period detection for the low-frequency signal. (b) The algorithm for the period detector.

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

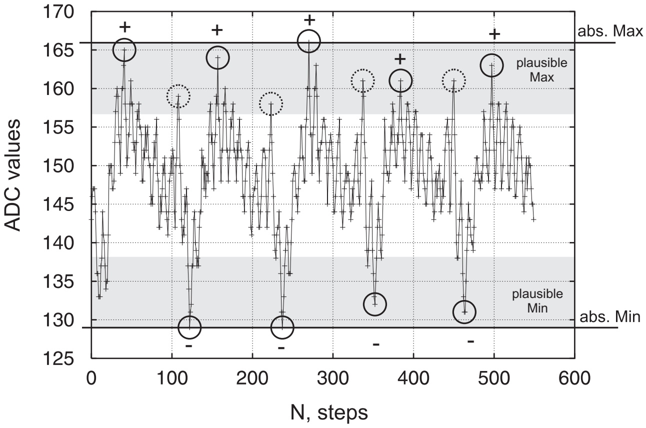


Fig. 6. The distorted signals (motors, power switchers, microcontroller noise) shown are constraints imposed on the peak detection.

Table 1
Constraints used in the CSP-solver for period calculation.

C	Constraint	Comment
C_1	$pl_j \in aMin + \Delta$, $ph_i \in aMax - \Delta$	Lower and upper peaks should lie within the corresponding plausibility regions
C_2	$ph{:}i > j$, $pl{:}j > i$	The index of the *Min* candidate should be greater than the index of the previous *Max*; the index of the *Max* candidate should be greater than the index of the previous *Min*
C_3	$ph, pl, ph, pl, \ldots$	*Min* and *Max* should follow one after the other
C_4	$ph - pl > 12$	To differentiate signal from noise

and stores the number of errors. When the number of errors is greater than a defined threshold, the filter updates the test criteria and thus adapts itself to the altered period. The threshold values are found experimentally; the best result being achieved when the threshold is equal to the depth of the averaging filter used in the control scheme. See Section 4.

3.2. Detection of blocked wheels

The odometric system possesses inertia of $P/2$, that is, equal to a distance of 18.06 mm. Since a microrobot uses a stepwise change of velocity, this inertia does not influence the odometric capabilities of the robot. However, this does not improve the detection of blocked wheels (wheels that do not rotate), which must be detected without delay. There are various methods for detecting such stoppages. Fig. 8 shows two plots, where the corresponding wheel is blocked in two different cases: in Fig. 8a only the motor board is running, in Fig. 8b both boards are running. Comparing Fig. 8 with, for example, Fig. 6, we can observe difference in the amplitude of the peaks between *Min* and *Max*. The normal difference should be more than 15 ADC values (achieved by initial mechanical calibration of the odometric system). When the difference is less than 15, this can point to stoppage of the wheel.

Another principle of stop detection uses the fact that the detector works sequentially on the left and the right wheels. When a wheel does not rotate, the corresponding CSP-conditions are not satisfied and the detector does not calculate the period P. When the detector does not deliver P within 3–5 cycles and $Max - Min > 15$, these conditions can also be used for reliable detection.

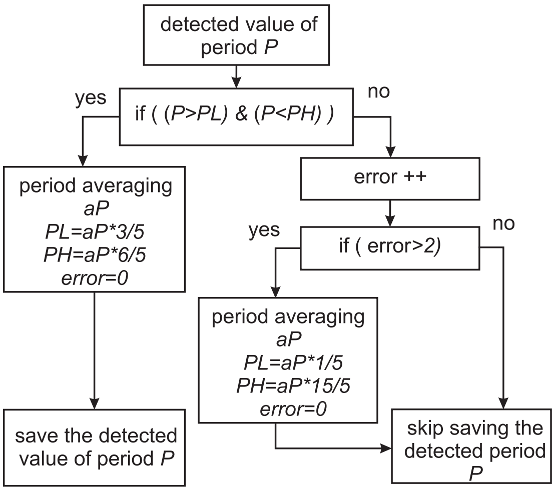


Fig. 7. Algorithms of the adaptive plausibility filter.

4. Auto-calibration of differential and absolute velocities

Fig. 9 shows an evolution of the detected period P for the first several minutes of a robot's motion. We observe that P undergoes several fluctuations, is different for both motors, and drifts when the motors become warm.

Period fluctuation is a slight variation of the period detection in different cycles. The underlying reason for this is electronic noise, which creates maximum close peaks higher than the true maximum (or minimum) of the signal. There is no way to remove this effect. Therefore, we create an averaging filter that smoothes this variation. However, this filter increases the inertia of the control system: the system reacts to a change of period with a delay defined by the averaging.

Different periods in motors primarily arise from differential mechanical friction in the internal and external motor gears and also from the friction of the wheels. In Fig. 9 we can see that the period of the right motor is smaller than that of the left. That is, the robot always turns left. To compensate for this difference, either the left motor should receive a longer PWM signal or the right motor a shorter one.

The motor drift is a typical nonlinear effect, which appears due to the initial warming of the DC motors. Within the first two to three minutes of motion their rotation frequency increases, sometimes by as much as 10–20%. Unfortunately, the motor drift is different for each robot (it also depends on friction in the transmission), so that the stabilization of a defined period must also be done by the control system.

For the control, we use a linear differential control scheme, shown in Fig. 10. The control scheme detects differences in wheel rotation and regulates corrections of the right and left PWM-control signal, as well as normalizing an absolute velocity. The correction values obtained are stored in the nonvolatile EEPROM memory. The auto-calibration procedure is relatively quick, see Section 5, and must be performed for all velocities available for the microrobot. This is also the reason why the robot has a set of 10 fixed calibrated velocities from $P = 90$ to $P = 160$. This initial correction should be made only once; after that the robot moves with normalized values and performs a small auto-correction when needed.

4.1. Systematic and non-systematic errors in the odometric system

The odometric system has several sources of systematic error. The first is the inertia of $P/2$ in the period detection. When the differential velocity is suddenly changed, the robot receives feedback only after traveling 18 mm. Usually, this error does not have an essential impact on behavior, because after 18 mm a robot always performs a correction of the differential velocity.

The second source of systematic error is the inaccuracy of peak detection, due to sampling and additive noise, see Fig. 11. This is

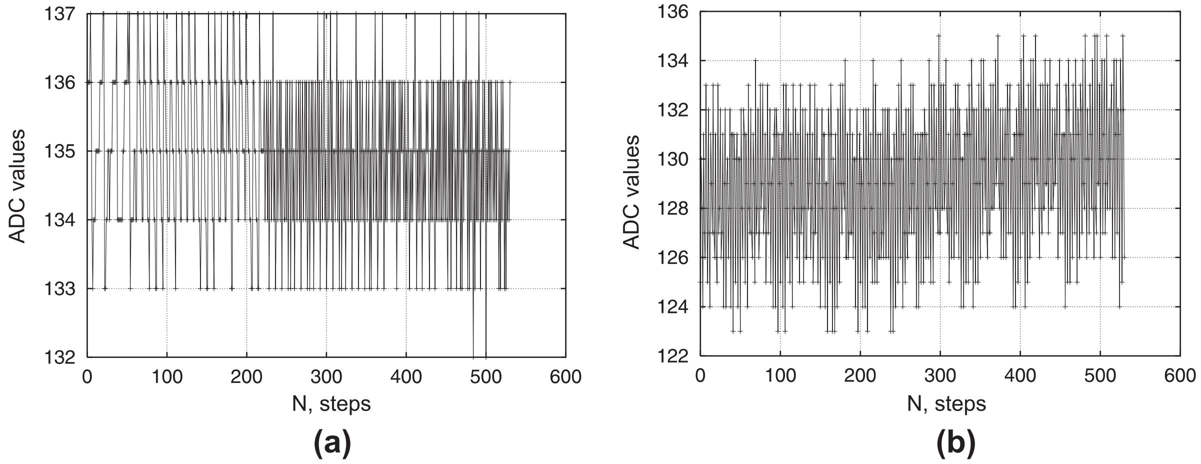


Fig. 8. ADC values from the odometric sensor when the corresponding wheel is blocked and does not rotate. (a) Only the motor board is running (microcontroller, H-bridges and motors) and (b) both boards are running (two microcontrollers, H-bridges, motors, power switchers of IR emitters).

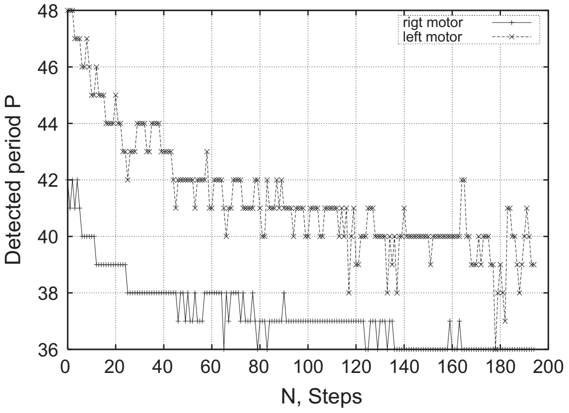


Fig. 9. Detected period *P* after the CSP-detector and the adaptive plausibility filter.

also related to *P*/2. However, it is valid only for short distances. Due to uniform distribution of sampling inaccuracy, the averaging filter removes this effect for long-distance movement and for velocity calibration. In only one case can this not be removed: when the *Min* peaks are extremely elongated and the peak detector has problems in determining the exact position of a peak. When this happens on one wheel, the robot almost always determines the rotation period of this wheel wrongly, and eventually slews left or right. Moreover, a robot cannot normalize the differential velocity. Since this seldom happens (fewer than 2% of all cases), we introduce a manual correction coefficient, which absorbs this constant inaccuracy. As an alternative approach to improve the signal–noise relationship, the surfaces of plastic gears can be slightly changed by grinding, as shown in Section 5.

The final source of systematic errors is the nonlinear mechanical friction in plastic gears. Due hand-assembling and the need to use glue, glue residues can remain on the gear, increasing the friction in that part. Usually this effect disappears over time, however, while it exists, it slows the rotation of one of the wheels. To compensate, we again use a manual correction coefficient. Based on preliminary tests (see Section 5), we estimate the maximum systematic error in the distance calculation (based on medium distance – 100 mm) about 6 mm – 6%, in the calculation of the rotation angle – about 10° (90° rotation) – 11%, in the stabilization of differential velocity – 0 (with the manual correction coefficient), in the stabilization of absolute velocity – 3 mm/s.

There is one typical non-systematic error for odometric systems, which do not receive motion feedback from the ground: the friction between the wheels and the ground surface. When a robot starts moving with a velocity of more than about 400 mm/s, the wheel turns 1–2 rotations without surface cohesion. As a result, the robot can rotate at a small random angle before moving

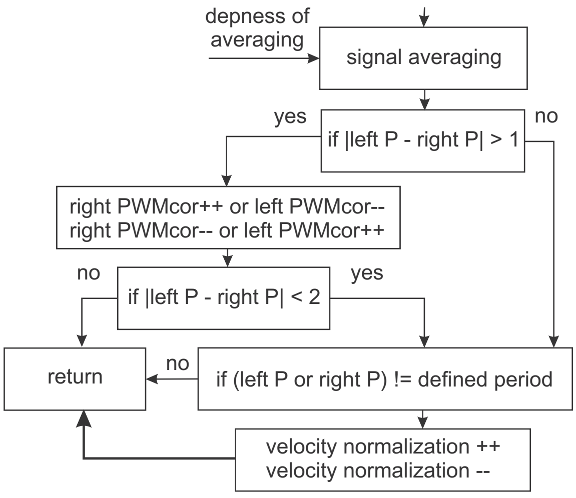

depness of averaging
signal averaging
yes
no
if |left P - right P| > 1
right PWMcor++ or left PWMcor--
right PWMcor-- or left PWMcor++
no
yes
if |left P - right P| < 2
return
no
if (left P or right P) != defined period
velocity normalization ++
velocity normalization --

Fig. 10. Structural scheme of auto-calibration of differential and absolute velocities.

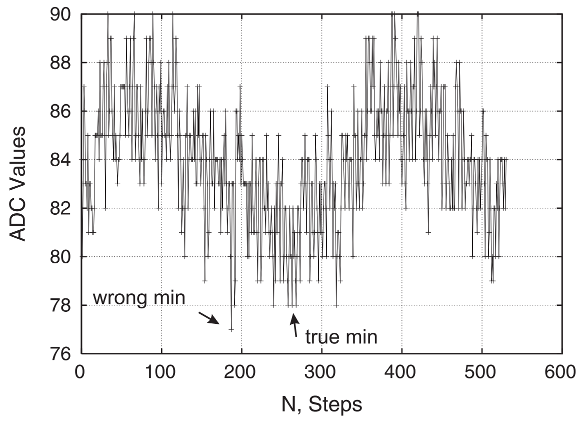


Fig. 11. Error in peak detection due to additive noise.

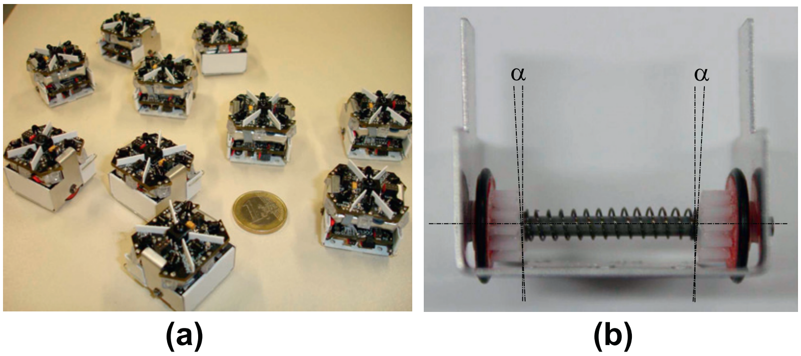
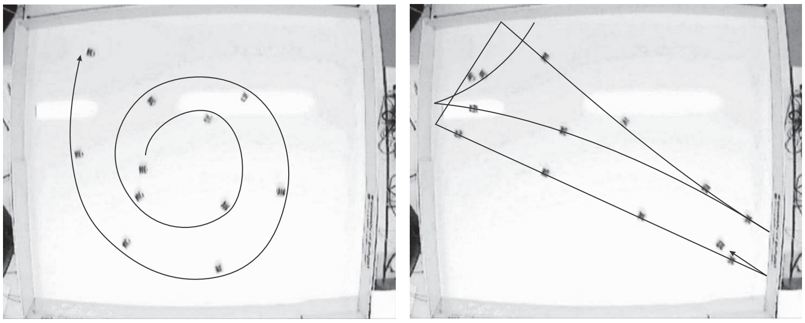


Fig. 12. (a) Ten randomly-chosen identical microrobots for test implementation and experiment. (b) Modifying the plastic gear to improve the level of signal obtained from the sensors, α varies between 2° and 4°.



(a) 1…11 sec (b) 12…24 sec

Fig. 13. Difference images of the motion of the first robot during the calibration phase (data for robot ID = 71). (a) Start of differential velocity calibration – the curved trajectory is readily visible. (b) Final differential velocity calibration; transition into straight forward motion. The last segment (straight line) of the trajectory corresponds to 1655 mm, which was covered in about 3 s (v = 551 mm/s).

straight forward. To remove this effect, especially when a defined rotation angle or distance traveled is required, the robot should start moving slowly.

5. Experiments

For our experiments, we randomly selected 10 of 107 currently available identical microrobots, see Fig. 12a. As the IR sensors are placed sideways on the motor PCB, the signal–noise relationship can be improved by grinding the surface of the plastic gears, as shown in Fig. 12b. During the revision of this work, we modified seven robots in this way. Three robots remained unmodified, to allow comparison of results between original and modified elements.

Table 2
Overview of signal parameters of robots and calibration times for large and small ΔP. The min. and max. values represent the corresponding ADC values of the signal from the IR sensors, averaged for 10 rotations of the wheels. The final three experiments were performed with unmodified gears. Robot ID = 64 has a manual correction coefficient of 3.

N	Rob. ID	R. wheel			L. wheel			Calibration time (s)	
		Max.	Min.	Δ	Max.	Min.	Δ	Large ΔP	Small ΔP
1	54	98	74	24	86	44	42	46	6
2	80	97	69	28	138	94	44	34	9
3	28	130	94	36	120	79	41	50	8
4	18	78	51	27	177	138	39	33	7
5	71	132	102	30	128	100	28	24	8
6	51	88	64	24	112	82	30	32	9
7	30	85	53	32	103	75	28	49	10
8	64	107	76	31	146	109	37	60	10
9	78	96	75	21	82	60	22	55	8
10	37	105	82	23	75	53	23	62	9

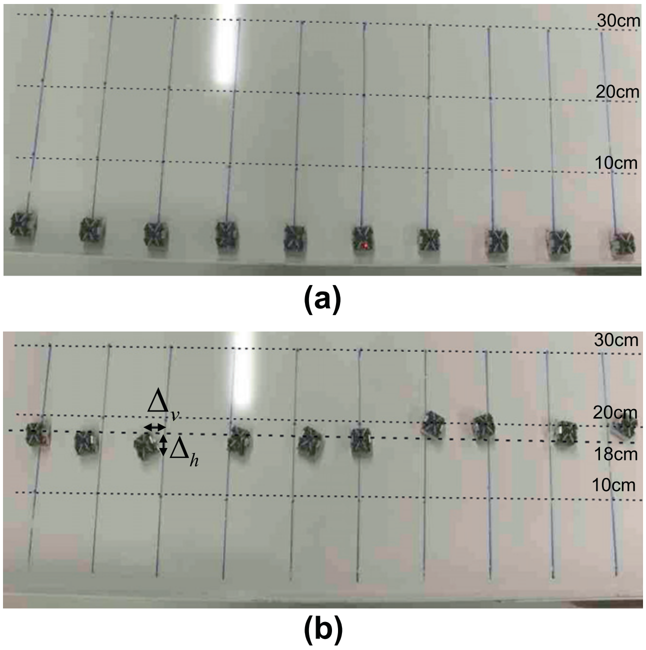


Fig. 14. One run of the experiment, measuring the distance traveled. (a) Initial position of robots. (b) Final position of robots after executing the command *move(180)*.

The experiments were designed to test three features of the odometric system: (1) the self-calibration time for large and small rotational differences between both wheels, (2) accuracy of distance measurement and (3) the odometric capability to work in the presence of a systematic inaccuracy, for example the moving of an object.

The idea behind the first experiment is explained in Fig. 9. Since the robots are assembled manually, the mechanical parameters of the right and left wheels differ. Thus, the initial period P and the motor drift caused by heating are different for both wheels and the odometry needs to calibrate the corresponding coefficients to allow the robot to move straight forward. In this experiment, all the robots were first calibrated, then had the values of their correction coefficients manually altered by 20; $\Delta P = 20$ introduces a large rotational difference. As shown in Fig. 13a, this results in the robot following a curved trajectory in the first phase of its movement. Then, the odometric system calibrates the differential velocity, see Fig. 13b, and both wheels move at the same rotational speed. This experiment was repeated with all 10 robots. The calibration time and the signal levels are shown in Table 2.

In addition, we measured the time needed for the correction of a small deviation, for example between cold and warm motors. For this, correction coefficients for warm motors were stored and used as the correction for cold motors during the initial phase of movement. Normally, $\Delta P = 3$–5 is small in this case.

As we see from these data, a difference between max. and min. values of less than 20 does not allow reliable discrimination between the signal from the IR sensors and system noise. The gears enabled the enlarging of the peak-to-peak amplitude of the signal. Moreover, the elongated region of the max. or min. values (that is with many similar maximal or minimal values of a signal), mentioned in Section 4, normally requires a manual correction

Table 3
Values of Δ_h and Δ_s (in mm) for three attempts.

N	Rob. *N*	Attempt 1		Attempt 2		Attempt 3		Δ_h		Δ_v	
		Δ_h	Δ_v	Δ_h	Δ_v	Δ_h	Δ_v	Mean	StDev	Mean	StDev
1	54	0	0	3	−10	0	2	1	1.7	−2.6	6.4
2	80	12	13	5	10	−5	−15	4	8.5	2.6	15.3
3	28	14	16	0	0	−6	20	2.6	10.2	12	10.5
4	18	5	−12	12	25	0	0	5.6	6.0	4.3	18.8
5	71	3	20	10	−18	−7	20	2	8.5	7.3	21.9
6	51	0	0	−5	10	12	−20	2.3	8.7	−3.3	15.2
7	30	−18	−5	15	−20	−10	20	−4.3	17.2	−1.6	20.2
8	64	−19	15	10	15	12	25	1	17.3	18.3	5.7
9	78	−15	−12	−15	−20	8	−15	−7.3	13.2	−15.6	4.0
10	37	−20	−15	17	−20	−10	20	−4.3	19.1	−5.0	21.7
						Overall data		0.2	10.9	1.6	15.8

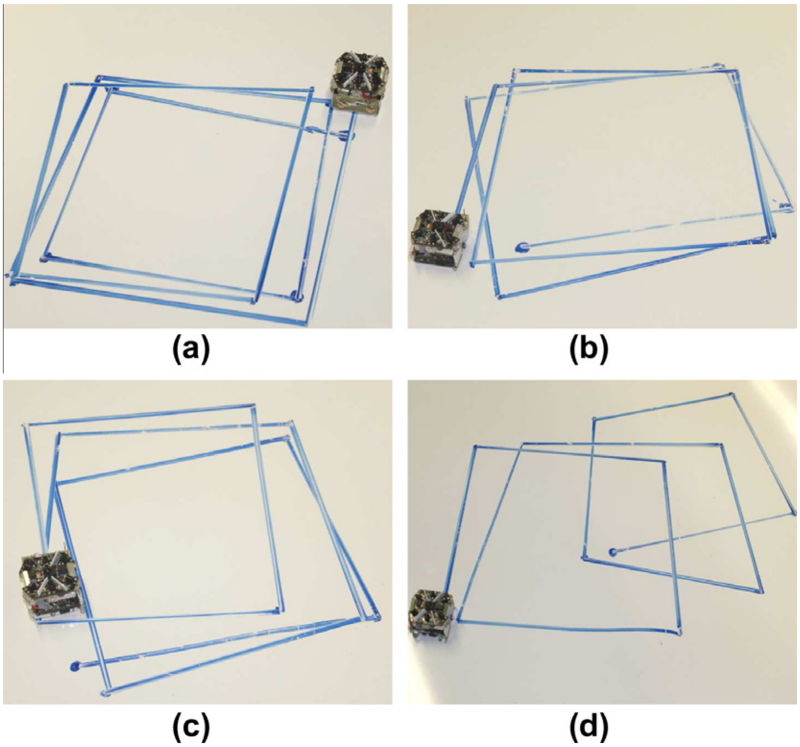

(a)
(b)
(c)
(d)

Fig. 15. Testing the accuracy of an odometric system by running a defined pattern. (a and b) Distance traveled is defined by the odometry and rotation angle is defined by the 3D-accelerometer. (c and d) Distance traveled and rotation angle are defined only by the odometric system.

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

coefficient, as for example, robot ID = 64. Modifying the gears enabled the avoidance of such flat regions on the gear surface, in turn avoiding manual corrections.

The second experiment represents a typical task for a robot: move a pre-defined distance. It reflects two parameters: the calibrated velocity v for each wheel and the distance measurement, made by accumulating the values of δ for each wheel. For this experiment, the main board transmitted the value of 180 as the target distance (in mm). The motor board multiplied this by a coefficient of 36, meaning the robot stops moving when the variable *goneDistance* $\geqslant 6480$ (we used the factor 10 in the coefficient to avoid calculations involving float variables). To increase accuracy, we used the second calibrated velocity with $P = 130$. For this we used $\delta = 0.2779$ mm, which corresponds to the value of distance traveled of 180 mm. Fig. 14 shows this experiment measuring the distance traveled. Ten robots were placed on the lines and we transmitted the command *move(180)* by remote control. Proximity sensing was switched off.

Table 3 shows two parameters: deviation Δ_h from the target distance of 180 mm and deviation Δ_v from the straight line, as shown in Fig. 14a. All the robots performed relatively well at measuring the distance traveled, the maximum single $\Delta_h = 20$ mm (11% inaccuracy), the maximum averaged $\Delta_h = 7.3$ mm (3.8% inaccuracy) and statistics for more than 30 trials with different robots $\Delta_h = 0.2$ mm. The main inaccuracy is related to low friction between the wheels and the surface of the arena, meaning the wheels turn without moving the robot at the first moment of movement. This has an impact on rotation. As can be seen in Table 3, Δ_v has larger Mean and StDev values than Δ_h. Since, during rotation, one wheel rotates back and the other forwards, inaccuracy of rotation due to low friction is relatively large.

To demonstrate this effect, and inspired by other tests of odometric systems in which a robot runs in a defined pattern [6], we reproduced one such experiment, as shown in Fig. 15. In this experiment, the robot must move through a square and in an ideal case, return into the point at which it began. In Fig. 15a and b we show two runs in which a robot uses a 3D-accelerometer to define the proper rotation angle and in Fig. 15c and d, runs in which it uses only the odometric system. In all cases, the distance traveled is measured only by odometry. In the first case, the distance between the initial and final points is between 5 and 40 mm; in the second case between 50 and 300 mm.

The final experiment underlies more complex approaches, with cooperative actuation by many robots. In this case, a robot moves through a half-square (170 mm direct, rotation left 90, 170 mm direct) and then moves an object for 25 mm. Object moving is included because this imposes a load on the locomotive system and stresses the odometry in calculating δ, see Section 3. The

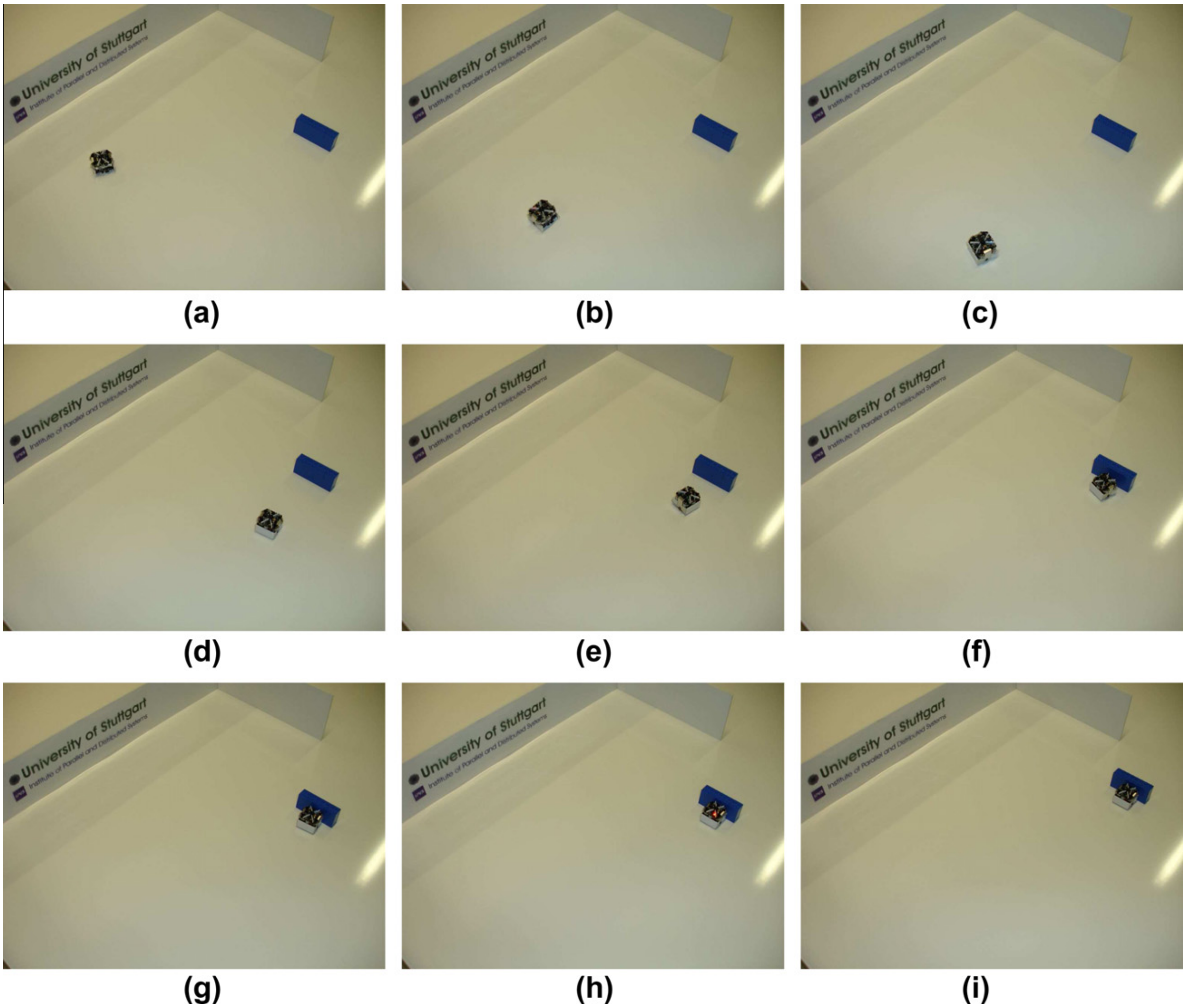


Fig. 16. Test of odometric system by moving an object. (a) Initial setup. (b–i) Images taken at 3 s intervals.

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004

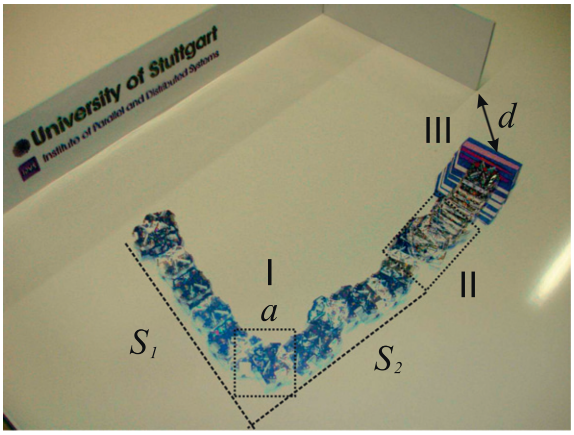


Fig. 17. Motion of one robot and moving of an object. This image is calculated as a difference image from 50 motion images. The numbers denote areas where the odometric system suffers systematic errors: I – area of small-angle rotation, II – scanning area, III – shifting area.

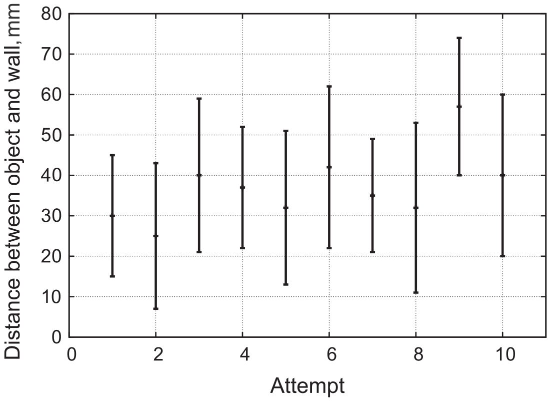


Fig. 18. Distance between object and the wall for each attempt shown in Fig. 17.

starting position of the robot and of the object are shown in Fig. 16a. One particular run is shown in Fig. 16; the differential image of the movement is made up of 50 motion images and is shown in Fig. 17. As may be seen in this figure, the odometric system undergoes several systematic errors in the moving and rotation portions, as well during scanning and moving of the object. By repeating this experiment several times with the same robot, we are able to estimate the quality of the odometric system in terms of variations in the final position of the object.

After moving through a half-square, the robot starts looking for an object by scanning through ±60° with its IR beam [20]. When it encounters an object, the robot rotates perpendicularly to it, moves until it touches the object, moves it 25 mm and then stops. After this, the robot is manually replaced at its starting position and the cycle repeated. For each attempt, we measured the position of the center of the object in relation to the edge of the perpendicular wall.

Of 100 attempts (10 robots making 10 trials each, with the robots calibrated from the first experiment), 5% failed to reach the object, and 11% failed to move it. The reason for this type of failure was chiefly the 90° rotation, where a robot rotated through the wrong angle and arrived at the edge of the object. We estimated the inaccuracy of this rotation, overall attempts, as 11% (and 30% as a maximum single inaccuracy). Thus, instead of moving the object, the robot rotated it. We counted this type of result as a failure. For the successful attempts (84%), the average distance between the object and the wall varied between 25 mm and 57 mm. Individual variations within each attempt also indicated ±20 mm. In Fig. 18 we plot average distances and variations for each attempt.

6. Conclusion

In this paper we have discussed the odometric system for a microrobot of $30 \times 30 \times 20$ mm^3. The odometry developed allows the stabilization of differential and absolute velocity for a robot that uses non-stabilized (not step) DC motors and hand-built force transmission based on plastic gears. Moreover, the system allows the measurement of the distance traveled with a typical accuracy of about 4% (middle distance range, maximum inaccuracy 11%) with a rotation of a typical accuracy of about 11% (for a single rotation $\leqslant 90°$, maximum inaccuracy 30%). The odometric system does not use specific encoders that are difficult (sometimes even impossible) to install in the microrobot, due to its size. Rather, all that is required are two small IR-reflecting sensors, soldered sideways on the motor PCB. To obtain motion feedback from the wheels, we use IR-light reflected from the motor–wheel transmission system. This approach can also be applied in other applications where size is a limiting factor.

The most difficult issue is real-time signal processing in a small microcontroller with limited memory. We demonstrated the main algorithms based on the CSP-based signal processing of the rotation-period detector, adaptive plausibility filter, stop-motion detector and velocity auto-calibration. Despite considerable electronic noise in the system, the signal processing is stable; of all tested robots, in none did the odometric system completely fail. However, the system does have several sources of inaccuracy, mostly of mechanical origin. In 25% of the robots a small (in the range 1–4) manual correction coefficient is needed. The hand-assembly of the robots also introduces systematic errors in the odometry, which likewise need manual correction in some robots. This procedure can be automated; when a robot moves along the wall, in trying to keep a constant distance from the wall, it performs a full self-calibration. This calibration can be avoided by grinding a small angle on the gear to improve the signal–noise ratio. Comparing the performance of modified and unmodified gears, we observed improved results in the modified gears.

The odometric system described is an important instrument for spatial coordination in large-scale robotic swarms. The microrobots can measure their distance from an object with an accuracy of a few mm (in the medium distance range) [18]. Spatial coordination between the robots can be built on a combination of the distance sensor, odometric system, 3D-accelerometer, and the color and touch sensors (for object classification and detection). As demonstrated in several tests, using local coordination mechanisms and this hardware combination allows coordinated activity for object transport and pattern generation. The average accuracy, for a distance of about 10 body lengths and one rotation of 90°, is about 1 body length of the robot. Since motion feedback is sensed from the wheels and not from the ground, this result is comparable to encoder-based solutions [15]. It cannot be used for precise object manipulation by a single robot, however it is accurate and reliable enough to be used in swarm experiments, where multiple robots execute their tasks in parallel. Based on preliminary experiments [10,28,33] we believe the accuracy can be improved by performing collective odometric tasks. The exploration of these approaches – and especially their scalability – will be the subject of further work.

References

[1] Antonelli G, Chiaverini S. Experimental odometry calibration of the mobile robot khepera ii based on the least-squares technique. In: Proceedings of the 2005 IEEE international conference robotics and automation; 2005. p. 1465–70.

[2] Benet G, Blanes F, Sim JE, Pérez P. Using infrared sensors for distance measurement in mobile robots. Robot Auton Syst 2002;40:255–66.

[3] Bodi M, Thenius R, Schmickl T, Crailsheim K. Robustness of two interacting robot swarms using the BEECLUST algorithm. In: Troch I, Breitenecker F, editors. MATHMOD 2009 – 6th Vienna international conference on mathematical modelling.

[4] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems. New York: Oxford University Press; 1999.

[5] Bonarini A, Matteucci M, Restelli M. Automatic error detection and reduction for an odometric sensor based on two optical mice. In: Proceedings of the 2005 IEEE international conference on robotics and automation, 2005. ICRA; 2005. p. 1675–80.

[6] Borenstein J, Feng L. UMBmark: a benchmark test for measuring odometry errors in mobile robots. In: Mobile Robots. p. 113–24.

[7] Caselles J. Exploration of embodiment in real microrobotic swarm. Master Thesis. University of Stuttgart, Germany; 2005.

[8] Flynn A. Combining sonar and infrared sensors for mobile robot navigation. Int J Robot Res 1988;7:5–14.

[9] I-Swarm. I-Swarm: intelligent small world autonomous robots for micro-manipulation. In: European union 6th framework programme project no. FP6-2002-IST-1; 2003–2007.

[10] Jiménez M. Cooperative actuation in a large robotic swarm. Master Thesis. University of Stuttgart, Germany; 2006.

[11] Kernbach S. Structural self-organization in multi-agents and multi-robotic systems. Berlin: Logos Verlag; 2008.

[12] Kernbach S, editor. Handbook of collective robotics: fundamentals and challenges. Singapore: Pan Stanford Publishing; 2011.

[13] Kernbach S. Heterogeneous self-assembling based on constraint satisfaction problem. In: Martinoli A, Mondada F, editors. 10th International symposium on distributed autonomous robotics systems, Springer tracts in advanced robotics. Springer; 2011.

[14] Kernbach S, Kernbach O. Collective energy homeostasis in a large-scale micro-robotic swarm. Robot Auton Syst 2011;59:1090–101.

[15] Kernbach S, Meister E, Schlachter F, Jebens K, Szymanski M, Liedke J, et al. Symbiotic robot organisms: REPLICATOR and SYMBRION projects. In: Proceedings of the 8th workshop on performance metrics for intelligent systems, PerMIS '08. New York (NY, USA): ACM; 2008. p. 62–9.

[16] Kernbach S, Nepomnyashchikh V, Kancheva T, Kernbach O. Specialization and generalization of robotic behavior in swarm energy foraging. Math Comput Modell Dyn Syst 2012;18:131–52.

[17] Kernbach S, Thenius R, Kernbach O, Schmickl T. Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. Adapt Behav 2009;17:237–59.

[18] Kornienko S, Kornienko O, Constantinescu C, Pradier M, Levi P. Cognitive micro-agents: individual and collective perception in microrobotic swarm. In: Proceedings of the IJCAI-05 workshop on agents in real-time and dynamic environments, Edinburgh, UK. p. 33–42.

[19] Kornienko S, Kornienko O, Levi P. Generation of desired emergent behavior in swarm of micro-robots. In: Lopez de Mantaras R, Saitta L, editors. Proc. of the 16th European conference on artificial intelligence (ECAI 2004). Valencia, Spain, Amsterdam: IOS Press; 2004. p. 239–43.

[20] Kornienko S, Kornienko O, Levi P, Minimalistic approach towards communication and perception in microrobotic swarms, in: Proceedings of the international conference on intelligent robots and systems (IROS-2005), Edmonton, Canada; 2005. p. 2228–34.

[21] Larsen T, Hansen K, Andersen N, Ravn O. Design of Kalman filters for mobile robots; evaluation of the kinematic and odometric approach. In: Proceedings of the 1999 IEEE international conference on control applications, vol. 2; 1999. p. 1021–6.

[22] Levi P, Kernbach S, editors. Symbiotic multi-robot organisms: reliability, adaptability, evolution. Springer Verlag; 2010.

[23] Liu W, Winfield A. Autonomous morphogenesis in self-assembling robots using IR-based sensing and local communications. In: Dorigo M, Birattari M, Di Caro G, Doursat R, Engelbrecht A, Floreano D, et al., editors. Swarm intelligence. Lecture notes in computer science, vol. 6234. Berlin/Heidelberg: Springer; 2010. p. 107–18.

[24] Liu W, Winfield A, Sa J. A macroscopic probabilistic model of adaptive foraging in swarm robotics systems. In: Proceedings of 6th Vienna international conference on mathematical modelling (Mathmod 2009); 2009.

[25] Mondada F, Bonani M, Raemy X, Pugh J, Cianci C, Klaptocz A, et al. The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions. p. 59–65.

[26] Moon TK, Stirling WC. Mathematical methods and algorithms for signal processing. Germany: Prentice Hall; 2000.

[27] Nolfi S, Floreano D. Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines. Massachusetts: MIT Press/Bradford Book; 2004.

[28] Prieto V. Development of cooperative behavioural patterns for swarm robotic scenarios. Master Thesis. University of Stuttgart, Germany; 2006.

[29] Sahin E. Swarm robotics: from sources of inspiration to domains of application. Heidelberg, Germany: Springer-Verlag; 2004.

[30] Schmickl T, Crailsheim K. Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. Auton Robots 2008;25:171–88.

[31] Scholz O, Dieguez A, Corradi P. Minimalistic large-scale micro-robotic systems. In: Kernbach S, editor. Handbook of collective robotics: fundamentals and challenges. Singapore: Pan Stanford Publishing; 2011. p. 517–41.

[32] Seyfried J, Szymanski M, Bender N, Estana R, Thiel M, Wörn H. The i-swarm project: intelligent small world autonomous robots for micro-manipulation. In: Swarm robotics. LNCS, vol. 3342. Springer; 2005. p. 70–83.

[33] Zetterström G. Collaborative actuation in microrobotic swarm based on collective decision making and surfacecolor identification. Master Thesis. University of Stuttgart, Germany; 2006.

Please cite this article in press as: Kernbach S. Encoder-free odometric system for autonomous microrobots. Mechatronics (2012), http://dx.doi.org/10.1016/j.mechatronics.2012.05.004