



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/149980/>

Version: Published Version

Article:

Trenkwalder, S., Esnaola, I., Kaszubowski Lopes, Y. et al. (2020) SwarmCom : an infra-red-based mobile ad-hoc network for severely constrained robots. *Autonomous Robots*, 44 (1). pp. 93-114. ISSN: 0929-5593

<https://doi.org/10.1007/s10514-019-09873-0>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



SwarmCom: an infra-red-based mobile ad-hoc network for severely constrained robots

Stefan M. Trenkwalder¹ · Iñaki Esnaola¹ · Yuri Kaszubowski Lopes^{2,3} · Andreas Kolling⁴ · Roderich Groß¹

Received: 31 August 2018 / Accepted: 22 June 2019
© The Author(s) 2019

Abstract

Swarm robotics investigates groups of relatively simple robots that use decentralized control to achieve a common goal. While the robots of many swarm systems communicate via optical links, the underlying channels and their impact on swarm performance are poorly understood. This paper models the optical channel of a widely used robotic platform, the e-puck. It proposes SwarmCom, a mobile ad-hoc network for mobile robots. SwarmCom has a detector that, with the help of the channel model, was designed to adapt to the environment and nearby robots. Experiments with groups of up to 30 physical e-pucks show that (i) SwarmCom outperforms the state-of-the-art infra-red communication software—libIrcom—in range (up to 3 times further), bit error rate (between 50 and 63% lower), or throughput (up to 8 times higher) and that (ii) the maximum number of communication channels per robot is relatively low, which limits the load per robot even for high-density swarms. Using channel coding, the bit error rate can be further reduced at the expense of throughput. SwarmCom could have profound implications for swarm robotics, contributing to system understanding and reproducibility, while paving the way for novel applications.

Keywords Swarm robotics · MANET · Infra-red communication · Channel model · e-puck · libIrcom

1 Introduction

Communication has been proven beneficial for multi-robot systems that cooperatively perform tasks (Balch and Arkin 1994). While there are many forms of communication, this

work focuses on *explicit* communication, in which arbitrary data can be exchanged.

In swarm robotics systems, communication is often used to enable decentralized control (Hauert et al. 2010; Schmickl et al. 2011; Rubenstein et al. 2014; Garattoni and Birattari 2018). However, designers of such systems face a number of challenges. On many platforms, computational resources are severely limited. For the platform considered in this work, for example, 8 kB of primary memory are available, and only a small fraction of this can be dedicated to communication, as the robot needs to perform other functions, including sensing, control, and actuation. As the number of

S.M. Trenkwalder is a recipient of a DOC Fellowship of the Austrian Academy of Sciences.

This is one of the several papers published in *Autonomous Robots* comprising Special Issue on Robot Communication Challenges: Real-World Problems, Systems, and Methods.

✉ Roderich Groß
r.gross@sheffield.ac.uk

Stefan M. Trenkwalder
s.trenkwalder@sheffield.ac.uk

Iñaki Esnaola
esnaola@sheffield.ac.uk

Yuri Kaszubowski Lopes
yurilopes@utfpr.edu.br

Andreas Kolling
akolling@irobot.com

- ¹ Department of Automatic Control and System Engineering, The University of Sheffield, Sheffield, UK
- ² Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, UK
- ³ Department of Software Engineering, Federal University of Technology – Paraná (UTFPR), Dois Vizinhos, Brazil
- ⁴ iRobot, Pasadena, USA

robots in a swarm can be large, their communication network should be decentralized and scalable, and therefore, given the aforementioned computational constraints, a robot may be unable to handle more than a few communication channels. Moreover, due to the robots' mobility, the system has to cope with changes in the topology of the communication network.

Mobile ad-hoc networks (MANETs) are networks of mobile devices, called nodes, which connect wirelessly without needing additional infrastructure. The nodes in the network are all end-points that can exchange messages with multi-hop routing. A wide range of MANETs have been investigated, from some offering elementary routing to others addressing all layers of the Internet protocol stack (Macker and Corson 1998; Chlamtac et al. 2003; Conti and Giordano 2014).

In swarm robotics, the merits of MANETs have been recognized. Two types of MANETs are commonly deployed—radio-based (Li et al. 2009; Tutuko and Nurmaini 2014) and optical (Di Caro et al. 2009; Gutiérrez et al. 2009a; Rubenstein et al. 2012) ones. The underlying technologies have their advantages and disadvantages:

- Radio-based communication systems are widespread and standardized, making them accessible, inexpensive, and compatible with other systems. In comparison, optical communication systems are still in early development, with a lack of standardized off-the-shelf solutions contributing to increased development time. However, they are becoming an alternative to radio-based systems, requiring less energy per bit and offering higher throughput (Anees and Bhatnagar 2015; Malik and Singh 2015; Khan 2017).
- Radio waves penetrate many objects, making it possible to reach more nodes over larger distances even in cluttered environments. Nevertheless, most radio-based systems have an upper limit on the number of nodes that can be connected to a single access point. For example, a Bluetooth (Class 3) master node can be connected with up to 7 slave nodes within a 3 m radius. The high robot densities found in some swarm robotics systems, with hundreds to thousands of nodes within a 3 m radius (Mondada et al. 2009; Rubenstein et al. 2014), exceed the capabilities of most wireless networks. Optical signals, on the other hand, are line-of-sight transmissions which are obstructed by objects, reducing the range and the number of channels. As we will show, this makes the communication system applicable to high-density scenarios, hence improving the network's scalability.
- Radio-based communication requires an antenna, the size of which depends on the carrier's wavelength. This

makes miniaturization challenging.¹ In comparison, optical systems offer good miniaturization potential (e.g., Hirschman et al. 1996), and are therefore typically found in swarms of centimeter-scale or sub-centimeter-scale robots (Seyfried et al. 2005; Rubenstein et al. 2012).

While the robots of many swarm systems communicate via optical links (Seyfried et al. 2005; Caprari and Siegwart 2005; Gutiérrez et al. 2008; Arvin et al. 2009; Rubenstein et al. 2014; McLurkin et al. 2014), the underlying channels and their impact on swarm performance are poorly understood.

In this paper, we propose SwarmCom, an optical (infra-red) MANET for severely constrained robots. It provides a framework that facilitates systematic studies of swarms and their communication channels and the design of reproducible behavior in swarms. This paper presents two contributions:

- A channel model that describes the infra-red signals transmitted and received on a widely used swarm robotics platform, the e-puck. This is the first such model for any swarm robotics platform. It informs the system designer about the physical layer characteristics for the robot communication. We explain in detail how to create such a model for the e-puck and demonstrate its impact on swarm communication.
- SwarmCom, a MANET for robotic swarms. SwarmCom allows groups of robots to communicate using on-off-modulated infra-red signals. It provides a dynamic detector that, with the help of the channel model, was designed to adapt to the environment and nearby robots. Moreover, SwarmCom provides channel coding, which reduces transmission errors, and a carrier sense multiple access (CSMA) protocol by which multiple robots compete for access. SwarmCom outperforms the state-of-the-art infra-red communication software for e-pucks—libIrcCom (Gutiérrez et al. 2009a)—with respect to communication range, bit error rate, and throughput. These improvements lead to a communication system that is better performing and more reliable, opening up new possibilities for research on miniature swarms.

The remainder of the paper is organized as follows. Section 2 describes the e-puck platform. Section 3 presents a channel model for the e-puck's infra-red communication hardware and validation experiments. Section 4 describes SwarmCom, including modulation and demodulation schemes, channel coding, and media access control. In Sects. 5 and 6, SwarmCom is evaluated in a series of experiments, and compared against libIrcCom. We first evaluate

¹ Although it is yet to be seen whether suitable for modern communication systems, it should be noted that antennas can be miniaturized to sub-micron scale (Chen et al. 2013).

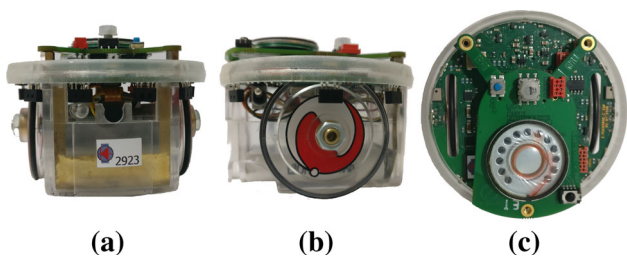


Fig. 1 An e-puck robot from different perspectives: **a** front, **b** side, and **c** top

SwarmCom on pairs of static robots, and then on groups of up to seven mobile robots. Section 7 presents the conclusions.

2 e-puck platform

The e-puck (Mondada et al. 2009) is one of the most widely used swarm robotics platforms (Cianci et al. 2006; Fischer and Hickinbotham 2010; Li et al. 2014; Chen et al. 2015; Nemeč et al. 2017). It was designed at the École Polytechnique Fédérale de Lausanne (Switzerland) and is available both commercially and under an open-hardware license.² The e-puck is shown in Fig. 1. It is cylindrical, with a diameter of 7 cm and a height of 5 cm. It has a differential wheel drive, enabling motion over flat terrain.

The e-puck contains a single microcontroller unit (MCU), a dsPIC30F6014a. This MCU offers 8 kB of RAM and is, therefore, a severely constrained device of class 1 (Bormann et al. 2014). The robot has a range of sensors, including eight infra-red proximity sensors, a directional camera, three microphones, and a three-dimensional accelerometer.

Each proximity sensor is a reflective optical sensor (Telefunken 1999). It has an emitter (infra-red LED) and a detector (phototransistor). To detect the proximity of nearby objects, first, the LED emits light. Any reflected light is then converted by the phototransistor into a voltage, which is provided to the MCU. A pair of proximity sensors can be used for communication.

The robot features Bluetooth capability, which it can use to share data with an external computer.

Figure 2 details the polar coordinates (r_i, θ_i) of proximity sensor $i \in \{1, 2, \dots, 8\}$ in the robot’s local coordinate system. The emitter and detector contained on a sensor are symmetrically arranged with an offset of ± 0.127 cm from the center. The coordinates of the emitter, $\mathbf{p}_{e,i}^{loc}$, and detector, $\mathbf{p}_{d,i}^{loc}$, of sensor i are given by:

$$\mathbf{p}_{e,i}^{loc} = \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \end{pmatrix} r_i + \begin{pmatrix} -\sin \theta_i \\ \cos \theta_i \end{pmatrix} 0.127, \tag{1}$$

$$\mathbf{p}_{d,i}^{loc} = \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \end{pmatrix} r_i + \begin{pmatrix} \sin \theta_i \\ -\cos \theta_i \end{pmatrix} 0.127, \tag{2}$$

² See www.e-puck.org.

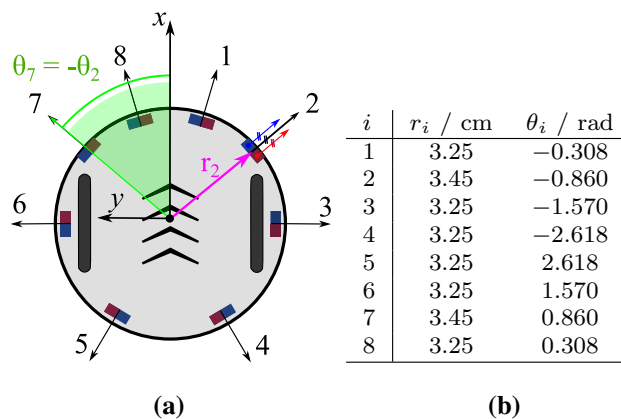


Fig. 2 Proximity sensors, with **a** their locations and orientations illustrated on the e-puck. The emitting (red) and sensing (blue) directions align with the sensor orientation. **b** Polar coordinates, (r_i, θ_i) , for sensor i (Color figure online)

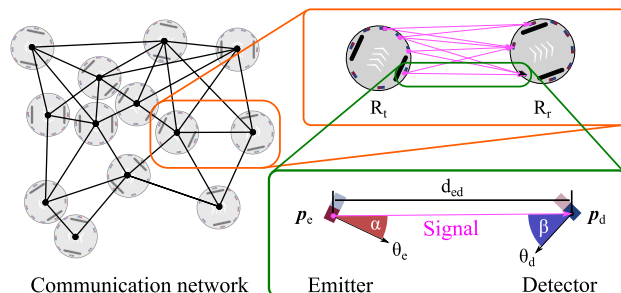


Fig. 3 The e-puck communication network (left) consists of links between pairs of neighboring robots. For each link (top right), the emitters of one robot can transmit a signal that is received by the detectors of the other robot. The signal propagates between emitter–detector pairs (bottom right). Due to occlusion, not all emitter–detector pairs are relevant

with $i \in \{1, 2, \dots, 8\}$.

3 e-puck channel model

This section presents a channel model that formally describes the infra-red signals transmitted and received by a swarm robotics system. The channel model can be used to form a communication link between a pair of robots. Moreover, it can be used to detect the proximity of nearby objects, where a signal is emitted, and its reflection detected. While the implementation of the model is specific to the e-puck platform, the underlying design process can be applied to other robots using infra-red based communication as well.

Figure 3 shows how a group of robots can form a communication network. Each node in this network is a robot that can have a link to other robots. As each robot emits/receives signals with multiple emitters/detectors, a robot-to-robot link is formed by multiple emitter–detector pairs.

The channel model characterizes:

- how a signal is transmitted between an emitter and a detector (emitter-to-detector model);
- how a signal is transmitted between two robots (robot-to-robot model);
- how a signal is transformed and represented in the receiving robot (measurement model).

3.1 Emitter-to-detector model

As a first step, characterising a communication system requires a formulation of the signal transmission and reception procedures. Consider a signal that is transmitted between an emitter–detector pair, as shown in Fig. 3. The emitter and detector are located at $\mathbf{p}_e, \mathbf{p}_d \in \mathbb{R}^2$ with orientation $\theta_e, \theta_d \in (-\pi, \pi]$, respectively.³ Then,

$$\alpha = \theta_e - \angle(\mathbf{p}_d - \mathbf{p}_e), \quad (3)$$

$$\beta = \angle(\mathbf{p}_e - \mathbf{p}_d) - \theta_d, \quad (4)$$

$$d_{ed} = \|\mathbf{p}_d - \mathbf{p}_e\|_2, \quad (5)$$

where α is the emission angle, β is the inclination angle, d_{ed} is the Euclidean distance between the emitter and detector, and $\angle(\mathbf{v})$ is the angle between vectors \mathbf{v} and $[1 \ 0]^T$.

Let the transmitted waveform $s(t) : \mathbb{R} \rightarrow \{0, 1\}$ for symbol $s \in \{s_0, s_1\}$ be an on–off modulated signal. The received signal intensity is

$$y(t) = h_c s(t) + n(t) \quad \text{with } 0 \leq t < T, \quad (6)$$

where T is the duration of the symbol, h_c is the channel attenuation coefficient, and $n(t)$ is zero mean additive white Gaussian noise (AWGN), which is a common assumption (Carruthers and Kahn 1997). The channel attenuation coefficient is the product of the signal attenuation caused by the emitter (h_e), the medium (h_m), and the detector (h_d):

$$h_c = h_c(\alpha, \beta, d_{ed}) = h_e(\alpha) h_m(d_{ed}) h_d(\beta). \quad (7)$$

Coefficients h_e , h_m , and h_d depend on variables α , d_{ed} , and β , respectively (see Fig. 3). In particular, the emitter coefficient, $h_e(\alpha) : (-\pi, \pi] \rightarrow [0, 1]$, models the attenuation caused by the emitter’s directionality. The maximum signal intensity is $h_e(0) = 1$ and, due to self-occlusion, $h_e(\alpha) = 0$ for all $\alpha \notin [-\frac{\pi}{2}, \frac{\pi}{2}]$.

The medium coefficient, $h_m(d_{ed}) : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$, models the attenuation of the signal when propagating through free space. We assume $h_m(0) = 1$ and $\lim_{d_{ed} \rightarrow \infty} h_m(d_{ed}) = 0$.

Similarly to $h_e(\alpha)$, the detector attenuation coefficient, $h_d(\beta) : (-\pi, \pi] \rightarrow [0, 1]$, models the attenuation caused

by the detector’s directionality, where $h_d(0) = 1$ and $h_d(\beta) = 0$ for all $\beta \notin [-\frac{\pi}{2}, \frac{\pi}{2}]$.

In Sect. 3.4, all model coefficients are determined.

3.2 Robot-to-robot model

The second step models the transmission of the signal from multiple emitters to multiple detectors (if present). Consider a signal that is transmitted between a pair of robots. The transmitting robot sends the signal with all its emitters. Each detector of the receiving robot measures the response, which is assumed to be instantaneous.

Let the transmitting robot be located at \mathbf{p}_t with orientation θ_t . Then, its emitter $i \in \{1, 2, \dots, 8\}$ is located at

$$\mathbf{p}_{e,i} = \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) \\ \sin(\theta_t) & \cos(\theta_t) \end{bmatrix} \mathbf{p}_{e,i}^{loc} + \mathbf{p}_t. \quad (8)$$

Similarly, let the receiving robot be located at \mathbf{p}_r with orientation θ_r . Then, its detector $j \in \{1, 2, \dots, 8\}$ is located at

$$\mathbf{p}_{d,j} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix} \mathbf{p}_{d,j}^{loc} + \mathbf{p}_r. \quad (9)$$

Based on (3)–(9), the received signal $y_j(t)$ at detector j is given by

$$y_j(t) = \min(1, \tilde{y}_j(t)), \quad (10)$$

$$\tilde{y}_j(t) = \left(\overbrace{\sum_i h_e(\alpha_{i,j}) h_m(d_{i,j}) h_d(\beta_{j,i})}^{h_c} \right) s(t) + n_j(t), \quad (11)$$

$$\alpha_{i,j} = \theta_i - \angle(\mathbf{p}_{d,j} - \mathbf{p}_{e,i}), \quad (12)$$

$$d_{i,j} = d_{j,i} = \|\mathbf{p}_{d,j} - \mathbf{p}_{e,i}\|_2, \quad (13)$$

$$\beta_{j,i} = \angle(\mathbf{p}_{e,i} - \mathbf{p}_{d,j}) - \theta_j, \quad (14)$$

where $n_j(t)$ is zero mean AWGN independent of the transmitted signal.

3.3 Measurement model

The third step models the signal demodulation and decision at the receiver. When the (optical) signal, $y_j(t)$, reaches detector j , a measuring circuit transforms it into an electrical signal, that is, a voltage.⁴ Figure 4 shows the measurement circuit. The incoming signal generates a proportional collector current $I_C(t)$ at time t . This causes a voltage drop from the supply voltage V_{cc} at resistor R , which results in the measured collector-emitter voltage

$$V_{CE}(t) = V_{cc} - R I_C(t). \quad (15)$$

³ If not mentioned otherwise, a global Cartesian coordinate system is assumed.

⁴ For simplicity, we drop index j throughout this section.

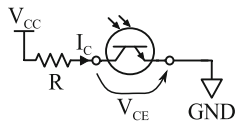


Fig. 4 Measuring circuit for an e-puck detector. The circuit is composed of a resistor, R , and phototransistor, which are connected in series to the robot's supply voltage, V_{cc} , and ground, GND

Therefore, $V_{CE}(t)$ is inversely proportional to the intensity of the incoming optical signal, $y(t)$.

Due to the superposition principle and saturation, the collector current is

$$I_C(t) = \begin{cases} I_{C,\infty}(t) & \tilde{I}_C(t) > I_{C,\infty}(t) \\ \tilde{I}_C(t) & \text{otherwise} \end{cases}, \quad (16)$$

$$\tilde{I}_C(t) = I_{C,S}(t) + I_{C,0}(t), \quad (17)$$

where $I_{C,\infty}$ is the maximum collector current (i.e., saturation current) of the transistor. $I_{C,S}$ and $I_{C,0}$ are the current components proportional to the signal and ambient light intensities, respectively. Voltage $V_{CE,max}(t) = V_{cc} - R I_{C,0}(t)$, caused by the ambient light component, is an upper bound. Similarly, voltage $V_{CE,min} = V_{cc} - R I_{C,\infty}$ is a lower bound, as $I_{C,\infty}$ is the maximum collector current. Note that $V_{CE,min}$ is the result of the circuit design and is assumed to be time-invariant.

The collector-emitter voltage, $V_{CE}(t) \in [V_{CE,min}, V_{CE,max}(t)]$, is sampled by an analog-to-digital converter (ADC). Let $\mathcal{M}_V : [0, V_{cc}] \rightarrow \mathbb{M}$ be the quantizer that maps $V_{CE}(t)$ to the measurement space $\mathbb{M} = \{0, 1, \dots, m_{sup}\}$, where $m_{sup} = 2^{n_{ADC}} - 1$ is determined by the ADC width (n_{ADC} bits).

Considering $V_{CE} \in [V_{CE,min}, V_{CE,max}(t)]$, the lower bound (saturation value), m_{min} , and upper bound (ambient light value), m_{max} , of the measurements are defined respectively as

$$m_{min} = \mathcal{M}_V(V_{CE,min}), \quad (18)$$

$$m_{max}[k] = \mathcal{M}_V(V_{CE,max}(t)). \quad (19)$$

The analog-to-digital conversion performs a linear discretization of time and value. Time-discrete values are indicated by $[\cdot]$ and $k = \lfloor \frac{t}{t_{sample}} \rfloor$, where t_{sample} is the sampling period.

Since $y(t) \propto I_C(t)$ and by applying (15)–(18), the infrared light conversion function $\mathcal{M} : \{[0, 1] \rightarrow \mathbb{M}\}$ maps the inverted light intensity, $1 - y(t)$, linearly to $[m_{min}, m_{max}]$ by

$$m[k] = \mathcal{M}(y(t)) \quad (20)$$

$$= \lfloor (m_{max}[k] - m_{min}) (1 - y(t)) + m_{min} \rfloor, \quad (21)$$

$$= \left\lfloor (m_{max}[k] - m_{min}) (1 - y^0(t)) + m_{min} + n_m(t) \right\rfloor, \quad (22)$$

where $y^0(t)$ is the noise-free received signal given by (10) when $n_j(t) = 0$ and $n_m(t)$ is zero-mean AWGN. Note that $m_{max}[k] = \mathcal{M}(0)$ and $m_{min} = \mathcal{M}(1)$ applies.

By combining the robot-to-robot model and the measurement model, the communication between two robots is fully described.

3.4 Model identification

The final step of modelling is to estimate the model parameters. To estimate h_e , h_m , h_d , m_{min} , m_{max} , and $n_m(t)$, a series of experiments is conducted.

3.4.1 Experiments with a single robot in ambient light

To characterize the ambient light value, m_{max} , and noise, $n_m(t)$, we conduct experiments with a single robot under different environmental conditions.⁵

In total, six experiments are conducted. In the first experiment, measurements are taken in the absence of ambient light—the robot is put in a closed box. In the second, third and fourth experiments, the robot is placed in the center or one of two corners of the experimental arena,⁶ the corners provide partial and complete shade, respectively. In the fifth experiment, the robot is put on a desk; in this situation, the absence of surrounding obstacles is assessed. In the sixth experiment, the robot is put close to a window (during daytime) to assess the influence of natural light. For each experiment, 2000 measurements per detector are taken.

Figure 5a shows the measurements' mean (\bar{m}_j) for each detector and experiment. The highest values (approx. 4095) were obtained in the closed-box experiment. Note that $4095 = 2^{12} - 1$ is the maximum value that the 12-bit ADC can provide. The lowest values (approx. 4075) were obtained in the experiment near the window.

To estimate m_{max} , we assume that the measurements, $m[k]$, follow a distribution obtained by discretizing the distribution $\mathcal{N}(m_{max}, \sigma^2)$, that is, $m_{max} + n_m(t)$, where $n_m(t) \sim \mathcal{N}(0, \sigma^2)$, and calculate the mean over all experiments, $m_{max} = 4080$. The ambient light thus causes the measurements, on average, to drop by 15, from 4095 to 4080.

Figure 5b shows the sample variances of measurements for each experiment. As the sample variances do not exceed 2.5, the noise is modeled as a (worst-case) white Gaussian process $n_m(t) \sim \mathcal{N}(0, 2.5)$.

3.4.2 Experiments with a single robot and light source

To characterize the saturation value, m_{min} , and detector attenuation, $h_d(\beta)$, we use a single robot and a light source (10 W

⁵ Note that the ambient light value, m_{max} , and noise, $n_m(t)$, are solely dependent on the robot's environment.

⁶ The experimental arena is described in Sect. 5.1.

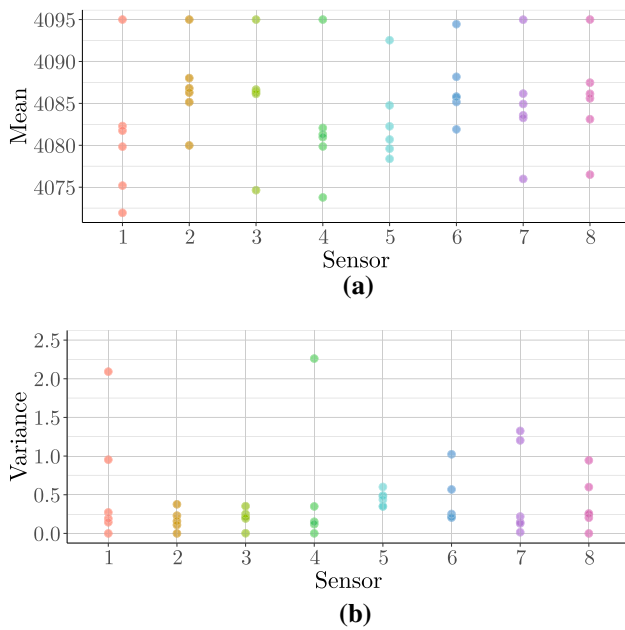


Fig. 5 Experiments where a robot is placed in one of six ambient light conditions while receiving with sensors 1, 2, ..., 8. **a** Mean and **b** variance of 2000 measurements per condition and sensor

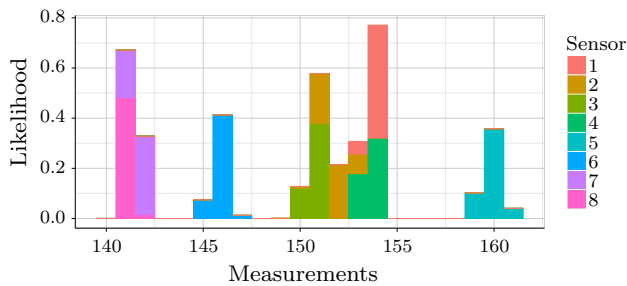


Fig. 6 Experiments where a light source is placed directly in front of the infra-red detector. Histogram of 2000 measurements per detector

LED). The light source is mounted at the height of the robot's detectors, and projects in parallel to the surface (floor). The light source is a flood light, providing a homogeneous and planar beam.

To estimate m_{min} , the robot is placed in front of the light source (2 cm gap), and oriented in one of eight configurations. In each configuration, one of the robot's detectors is directly oriented towards the light source. From this detector, 2000 measurements are taken.

Figure 6 shows the measurement distribution (i.e., $\mathcal{N}(m_{min}, \sigma^2)$) for each detector. The mean over all experiments is 149.80, which is rounded to 150 because the robots can only represent integers.

To identify $h_d(\beta)$, the robot is placed 10 cm from the light source with an orientation $\phi_k \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ relative to it. For each configuration, 2000 measurements per detector are taken.

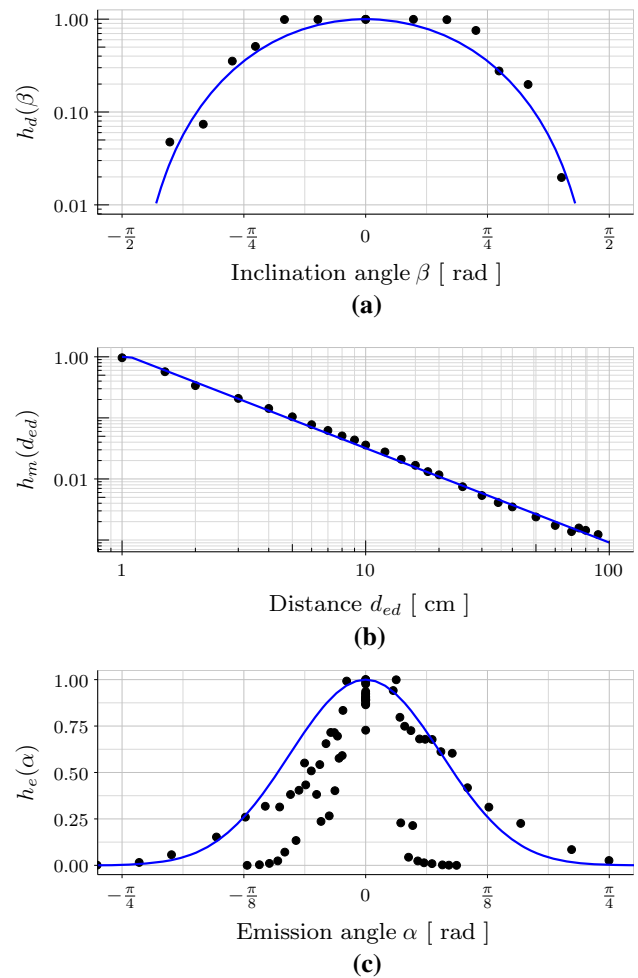


Fig. 7 Signal attenuation **a** at the detector, **b** through the medium, and **c** at the emitter. The dots show mean values derived from measurements with an inclination angle β , **b** emitter–detector distance d_{ed} , and **c** emission angle, α . The values are approximated by (24), (25), and (27), respectively

Figure 7a shows the attenuation at the detector, $h_d(\beta)$. Due to the homogeneous and planar light, we assume $h_e(\alpha) \cdot h_m(d_{ed}) = 1$, and that any attenuation of the signal intensity, $y = 1$, is caused by $h_d(\beta)$. As a result,

$$h_d(\beta_{j,k}) \approx \bar{y}_{j,k} = \mathcal{M}^{-1}(\bar{m}_{j,k}), \quad (23)$$

where $\bar{m}_{j,k}$ is the mean of the measurements taken by detector j in configuration k , $\beta_{j,k} = \theta_j - \phi_k$, and θ_j is the polar angle of detector j (see Fig. 2). The detector attenuation coefficient is approximated by

$$h_d(\beta) = |\cos(\beta)|^3. \quad (24)$$

3.4.3 Experiments with two robots

To characterize the medium attenuation, $h_m(d)$, and the emitter attenuation, $h_e(\alpha)$, we use two robots that are placed with

Table 1 Model parameters

n_{ADC}	12	Bit length for conversion
k_m	-1.54557	Medium attenuation decay
o_m	1.12202	Medium attenuation offset
m_{max}	4080	Ambient light value
m_{min}	150	Saturation value

an inter-robot distance⁷ of {7, 7.5, 8, 8.5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, 23, 25, 27, 32, 37, 42, 47, 57, 67, 77, 87, 97, 107} cm. Both robots are oriented in such a way that their sensors 1 are facing each other (i.e., $\alpha = \beta = 0$ for sensor 1). One robot is sending a constant signal with the emitter of sensor 1 while the other robot is measuring with all its detectors. For each sensor and location, 450 measurements were taken.

Figure 7b shows $\bar{y}_{j,k}$ for sensor 1 in relation to the emitter-detector distance, d_{ed} . As sensors 1 of both robots are aligned, we assume $h_e(\alpha) \cdot h_d(\beta) = 1$, and that any attenuation of the signal intensity, $y = 1$, is caused by $h_m(d_{ed})$.

The medium attenuation is approximated by

$$h_m(d_{ed}) = \begin{cases} 1 & d_{ed} \leq o_m^{-\frac{1}{k_m}} \\ o_m (d_{ed})^{k_m} & \text{otherwise} \end{cases}, \quad (25)$$

where the parameters are $k_m = -1.54557$ and $o_m = 1.12202$ (see blue line in Fig. 7b).

To estimate the emitter attenuation, $h_e(\alpha)$, any non-occluded detector is used (i.e., $|\beta_{j,k}| < \frac{\pi}{2}$). As $\bar{y}_{j,k}$ incorporates all attenuation components, it follows that

$$h_e(\alpha_{j,k}) \approx y'_{j,k} = \frac{\bar{y}_{j,k}}{h_m(d_k)h_d(\beta_{j,k})}. \quad (26)$$

Figure 7 shows the estimated light intensity obtained by each detector j in configuration k . Note that the signal intensity can differ significantly due to model inaccuracies of $h_d(\beta)$ when $\beta \approx \frac{\pi}{2}$. We approximate the emitter attenuation by

$$h_e(\alpha) = |\cos(\alpha)|^7. \quad (27)$$

In summary, Table 1 lists the model parameters.

3.5 Model evaluation

While in the previous experiments a robot was transmitting with only one emitter, an additional experiment is conducted in which a pair of robots transmit/receive with all

⁷ Note that the inter-robot distance is measured from and to a center of a robot. This results in a minimum of 7 cm (diameter of the robot).

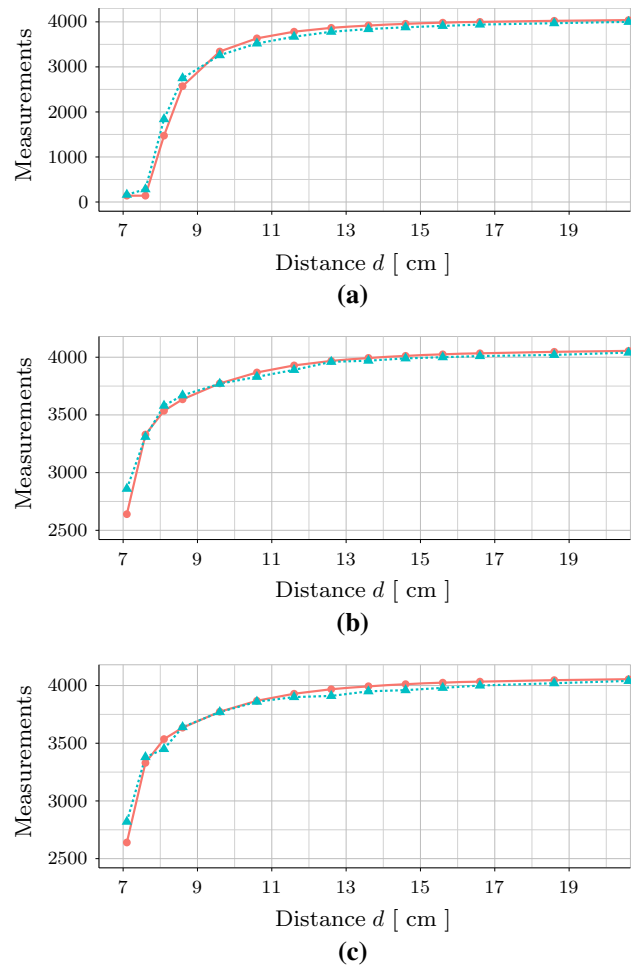


Fig. 8 Model predictions (red) and measurements (blue) from an experiment where two static robots, at distance d , exchange information using all their emitter-detector pairs: **a** sensor 1, **b** sensor 2, and **c** sensor 7 (see Fig. 2). Mean values based on 500 measurements. The range of axes was chosen to highlight interesting regions (Color figure online)

emitters/detectors. This causes the superposition of multiple signals from different angles and with different intensities. Furthermore, sensors 1 of the robots are no longer aligned.

The experiment is performed by placing two robots, facing each other, at a distance, $d \in \{7, 7.5, 8, 8.5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 21, 23, 25, 27, 32, 37, 42, 47, 57, 67, 77, 87, 97, 107\}$ cm. While one robot is sending a constant signal with all emitters, the other is receiving it with all detectors and transmits the measurements to a computer via Bluetooth. At each location, 500 measurements are taken.

Figure 8 shows the measurement means alongside the model predictions for three detectors. As the predictions closely match the measurements, the model parameters of Table 1 are used in the following sections.

4 SwarmCom: design and implementation

In this section, we describe SwarmCom, an optical MANET for severely constrained robots. SwarmCom enables a group of robots to establish a communication network dynamically. Within this network, pairs of robots transmit information via the infra-red channel, which was modelled in Sect. 3.

SwarmCom consists of:

- a *modulation* scheme to transmit data;
- a *demodulation* scheme to receive data;
- *channel coding* to detect and/or remove errors in the data that were introduced by the channel; and
- *medium access control* to manage the resource allocation.

4.1 Modulation

To convey information between a pair of robots, a signal is modulated onto the medium—the optical infra-red channel. The robot’s infra-red LEDs can only be switched on or off. Consequently, on–off–keying (OOK) is used as the modulation scheme. OOK is a binary modulation scheme that maps the information to be transmitted to symbols s_0 and s_1 . Symbols s_0 and s_1 are represented by the absence and presence of a carrier (infra-red light) over a certain duration, respectively. As a consequence, channels in idle state are assumed to output a s_0 symbol at each time step defined by the sampling period. A prefix consisting of an s_1 symbol is used to identify the beginning of data transmission. By default, data is divided into blocks of 15 bits in SwarmCom.

4.2 Demodulation

With the modulated signal conveying data, the demodulation aims to extract the sufficient statistics from the received signal and to decide the sequence of symbols that have been transmitted. The algorithm that realizes this is hereafter referred to as the detector. The maximum likelihood decision rule, $\Upsilon(m) : \mathbb{M} \rightarrow \{s_0, s_1\}$, does so optimally:

$$\Upsilon(m) = \begin{cases} s_0, & \text{if } P(S = s_0 | M = m) \geq P(S = s_1 | M = m) \\ s_1, & \text{otherwise} \end{cases}, \quad (28)$$

where $P(S = s | M = m)$ denotes the probability of symbol s being transmitted given that measurement m is observed.

4.2.1 Assumptions

To calculate $P(S = s | M = m)$, first, assumptions about the signal structure and robots’ positions are formulated.

We assume that any sequence of symbols is equally likely. Therefore, the a-priori probability of obtaining a symbol is

$$P(S = s_0) = P(S = s_1) = \frac{1}{2}. \quad (29)$$

Let the receiving robot be located at the origin of the coordinate system in an unbound environment and oriented towards the x-axis. The relative position of the transmitting robot is given in polar coordinates by random variable $L = (D, \Theta)$. Distance D is chosen from a gamma distribution (Forbes et al. 2010), $D \sim \mathcal{G}(k, \varphi)$, with shape parameter $k = 0.045$ and scale parameter $\varphi = 2.5$. Polar angle Θ is chosen from a uniform distribution with a probability density function, $f(\theta) = \frac{1}{2\pi}$ for all $\theta \in (-\pi, \pi]$.

Whereas the receiver robot’s orientation is embedded in Θ , the orientation of the transmitting robot, O , is chosen from a uniform distribution with a probability density function, $f(o) = \frac{1}{2\pi}$ for all $o \in (-\pi, \pi]$.

4.2.2 Received signal probability mass function

The signal probability mass function, referred to as $P(M = m | S = s)$, defines the likelihood of obtaining measurement m given that symbol s is sent, where $M \in \mathbb{M}$ and $S \in \{s_0, s_1\}$ are random variables.

In the absence of the carrier ($S = s_0$), only the ambient light and noise are present. Therefore, to compute $P(M = m | S = s_0)$, we exploit that M given $S = s_0$ follows $\mathcal{N}(m_{max}, \sigma^2)$.

To estimate $P(M = m | S = s_1)$, each 1000-, 720-, and 720-quantile of the aforementioned D , Θ , and O is applied to our channel model resulting in 5.184×10^8 synthetic observations per sensor. From these observations, the relative likelihood is calculated. Then, it is convolved with $\mathcal{N}(0, \sigma^2)$ to integrate noise.

Figure 9a shows $P(M = m | S = s_1)$ for each of the eight sensors. As most of a robot’s sensors will not point towards the emitting source, the measurement values for s_1 are similarly distributed than the measurement values for s_0 (see Fig. 9b). To reduce the similarity, the $\min\{\cdot\}$ operation is applied to all eight sensor measurements, choosing the smallest one, which originates from the sensor registering the highest signal intensity. This operation improves the signal distribution (see dotted lines in Fig. 9a and b) with small computational overhead.

4.2.3 Decision rule

With the estimated signal probability mass functions, the decision rule in (28) can be computed. To avoid computationally intensive probability calculations, (28) is simplified to

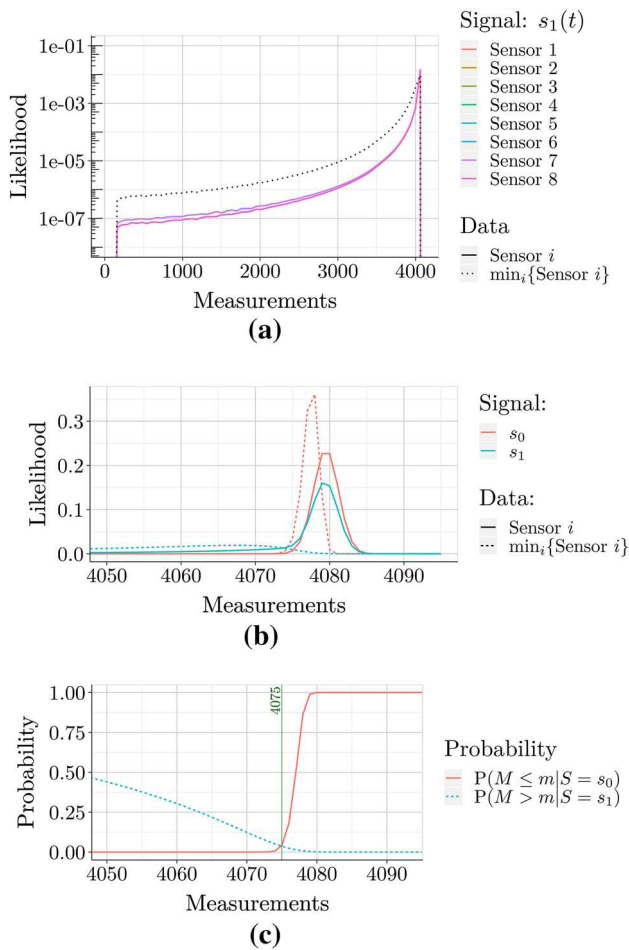


Fig. 9 Measurement statistics, produced by the channel model, for each sensor i of a robot that receives a signal from another robot in the environment. Symbols s_0 and s_1 correspond to the absence and presence of the carrier, respectively. **a** Distribution of measurements for s_1 ; **b** distributions of measurements for s_0 and s_1 , prior to (solid) and after (dotted) applying the min operator; **c** decision threshold m_t to discriminate between s_0 and s_1

$$\Upsilon(m) = \begin{cases} s_0, & \text{if } m \geq m_t \\ s_1, & \text{otherwise} \end{cases}, \quad (30)$$

where m_t , the *decision threshold*, is defined as

$$m_t = \max_{m \in \mathbb{M}} \{m \mid P(S = s_0 | M \leq m) \leq P(S = s_1 | M > m)\}. \quad (31)$$

With Bayes' theorem and (29), (31) is transformed into

$$m_t = \max_{m \in \mathbb{M}} \{m \mid P(M \leq m | S = s_0) \leq P(M > m | S = s_1)\}. \quad (32)$$

When applying (32) to the data of Sect. 4.2.2, the *decision threshold* is set to $m_t = 4075$ as shown in Fig. 9c. Furthermore, let the *relative decision threshold* be

$$m_{t,r} = m_{max} - m_t. \quad (33)$$

For $m_t = 4075$, we have $m_{t,r} = 5$.

4.3 Channel coding

While the demodulation aims to minimize errors for each symbol, channel coding can be used to detect and correct further errors in a *message* (i.e., a group of symbols) (Glover and Grant 2010). It relates to the noisy-channel coding theorem (Lapidoth 2009) that states that, for a given level of noise, the probability of error in the data can be made arbitrarily small, as long as the communication rate is below the Shannon capacity.

Whereas encoding methods are often computationally inexpensive, the computational requirements of decoding methods vary significantly. For instance, convolutional codes require probabilistic decoding (e.g., Viterbi decoder) which on severely constrained robots is not practical. In contrast, algebraic codes, such as binary block codes, require less computational expensive decoding through matrix or polynomial multiplications.

SwarmCom uses binary $[n, k]$ BCH codes (Glover and Grant 2010).⁸ These codes organize messages into n -bit long blocks that contain k bits of data. With an increasing number of parity bits, $n - k$, more errors can be corrected, but this reduces the effective throughput by $\frac{k}{n}$. SwarmCom implements $[15, 15]$, $[15, 11]$, and $[15, 1]$ BCH codes, which can correct up to 0, 1, and 7 errors, respectively. Depending on the choice of code, either the throughput (i.e., $[15, 15]$) or reliability (i.e., $[15, 1]$) is prioritized.

The matrices and their generator polynomial are available in the supplementary material website (Trenkwalder et al. 2018).

4.4 Medium access control

When multiple robots share the same medium and try to communicate, their access to the medium requires control to prevent the interference of messages (Guerroui et al. 2014; Hussain et al. 2017).

There are two forms of medium access control (Sesay et al. 2004)—controlled and random access. Controlled medium access divides the medium according to its physical properties (e.g., frequency or time) (Hadded et al. 2015; Glisic and Leppänen 2013). As every transmission has a dedicated slot, this access control technique is efficient for high network loads like streaming. However, as the number of robots within a swarm can be large, dividing the medium into a large number of frequency bands is often not feasible. Similarly, assigning a time-interval to each robot can cause long

⁸ The modular design of SwarmCom allows new codes to be added.

delays. As the delay increases with the number of robots, this method is not scalable. Furthermore, to allow the joining/leaving of robots, the assignment/removal of additional slots is commonly done by a central managing unit.

Random medium access allows robots to compete for access. This competition is performed through local information. Therefore, these methods are often decentralized, and work independently of the number of robots. Note that at high network loads, this method can cause a large number of collisions. Therefore, its performance would be reduced.

As many swarm robotics systems use infrequent communication, SwarmCom provides carrier sense multiple access (CSMA) (Shi et al. 2013; Wang et al. 2017) —a random medium access control method. A robot listens to the channel before sending a message. If the channel is free, the message is transmitted. Otherwise, the robot waits until the channel is free and then repeats the process. As a result, CSMA is decentralized as the robot uses only local information. This method is also scalable as the sensing of the channel state depends solely on the network load and not on how many robots are listening.

4.5 Implementation

SwarmCom has been implemented on e-pucks running the open-source OpenSwarm operating system (Trenkwalder et al. 2016). It uses a time-multiplexed on-board ADC to sample the voltages of the eight detectors, where bit detection is performed for every sample. Furthermore, the detector determines the ambient light value, m_{max} , during the start-up of the robot and adapts m_{max} during run-time to compensate for ambient light changes, as detailed in the following sections.

SwarmCom will be made available under an open-source license in the future. It is currently integrated within the OpenSwarm operating system, which is available at <https://github.com/OpenSwarm/OpenSwarm/tree/com-experiments>.

5 SwarmCom: static robots evaluation

In the following, we validate the SwarmCom implementation. This section reports on experiments involving a pair of static robots. One robot, R_t , transmits messages with all its emitters, while the other robot, R_r , receives them with all its detectors. SwarmCom performs modulation and demodulation. Unless otherwise stated, no channel coding is used.

5.1 Experimental setup

The experiments are conducted in an indoor arena of size 400 cm × 300 cm. The arena has a grey floor that is surrounded by 50 cm tall white walls. It is illuminated from

above by fluorescent light tubes; natural light is prevented from entering the room.

To automate the experiments, we use an external computer that is connected to each robot via a dedicated Bluetooth link.⁹ The experimental process is as follows:

- The external computer generates a message containing 1 to 5 blocks of data; each block has 15 bits. The number of blocks and their contents are chosen randomly from uniform distributions.
- The message is transmitted to R_t using Bluetooth. If no acknowledgment is received within 500 ms, it is retransmitted without affecting the result.
- Once the message has been received, R_t echoes it back to the computer as an acknowledgment, and, at the same time, starts the transmission to R_r .
- Once R_r has received the message from R_t , R_r transmits it to the computer.
- The computer marks a message as lost if its receipt was acknowledged by R_t , but after 1 s no message was received from R_r .

In the experiments, robots R_t and R_r are placed in the middle of the arena, oriented towards each other. Initially, their centers are 7 cm apart (the robots are touching each other). Above experimental process is executed repetitively. After obtaining 100 successful messages at a location, R_r moves away from R_t by 1 cm. If R_r has moved by 100 cm or 10 consecutive message transmissions have been marked as lost, the process is aborted. In the latter case, we assume that the end of the communication range has been reached.

The following performance measures are reported:

- The *communication range* is the furthest distance between two robots in which 100 messages were received.
- The *transmission time* is the time difference between receiving acknowledgments from R_t and R_r , respectively.
- The *bit rate* is the number of transmitted bits divided by the transmission time.
- The *bit error probability*, P_e , is estimated as the Hamming distance (i.e., number of altered bits) between data from R_t and R_r divided by the bit length of the message.
- The *probability of error-free transmission*, P_f , for an n -bit message is

⁹ The experiments focus on the quality of the optical infra-red communication channel between the robots. For the Bluetooth communication with the computer, a simple checksum is used. If errors are detected, the corresponding messages are retransmitted.

$$P_f(c) = \sum_{i=0}^c \binom{n}{i} P_e^i (1 - P_e)^{n-i}, \tag{34}$$

where c is the maximum number of correctable errors (due to channel coding). In other words, P_f is the probability that no more than c bit-errors occur. As the error correcting code can correct up to c errors, with probability P_f the transmission can be considered error-free.

– The probability of message loss, P_l , is

$$P_l = \frac{N_l}{N}, \tag{35}$$

where N_l and N are the number of lost messages and the number of messages, respectively.

5.2 Experiments with a fixed decision threshold

We examine the impact of using a fixed relative decision threshold, $m_{t,r} \in \{5, 10, 25, 50, 100, 250, 500, 750, 1000, 2000\}$. We test three different bit rates: 310 bps (the lowest possible bit rate offered by the hardware), 650 bps, and 1160 bps.

Figure 10 shows that the bit error probability is high when using a fixed decision threshold. The larger the threshold, the lower the bit error probability gets. However, increasing the threshold results in a reduction of sensitivity and hence, of the communication range.

It is worth noting that the bit error probability does not always monotonically increase with the inter-robot distance, especially when a high bit rate is used. A possible reason could be the measuring circuit. Short distances result in high signal intensities that cause high charges of capacitive elements. Small discharging currents could cause slow changes from one symbol to another and, hence, cause inter-symbol interference.

5.3 Experiments with a dynamic decision threshold

The detection was changed to adapt the threshold depending on the received signal intensity dynamically. In the following, m_t and $m_{t,r}$ refer to the default thresholds, as previously used in Sect. 4.2.3. For a signal to be detected, its amplitude, measurement A , has to be lower than the original threshold, m_t [see (30)]. If this is the case, amplitude A is then used to calculate a dynamic decision threshold,¹⁰

$$m_{t,dyn} = \left\lfloor \frac{|m_t + A|}{2} \right\rfloor = \left\lfloor \frac{|m_{max} - m_{t,r} + A|}{2} \right\rfloor. \tag{36}$$

After receiving a message, the dynamic decision threshold is reset to the default value (m_t).

¹⁰ Note that the definition ensures $m_{t,dyn} < m_t$.

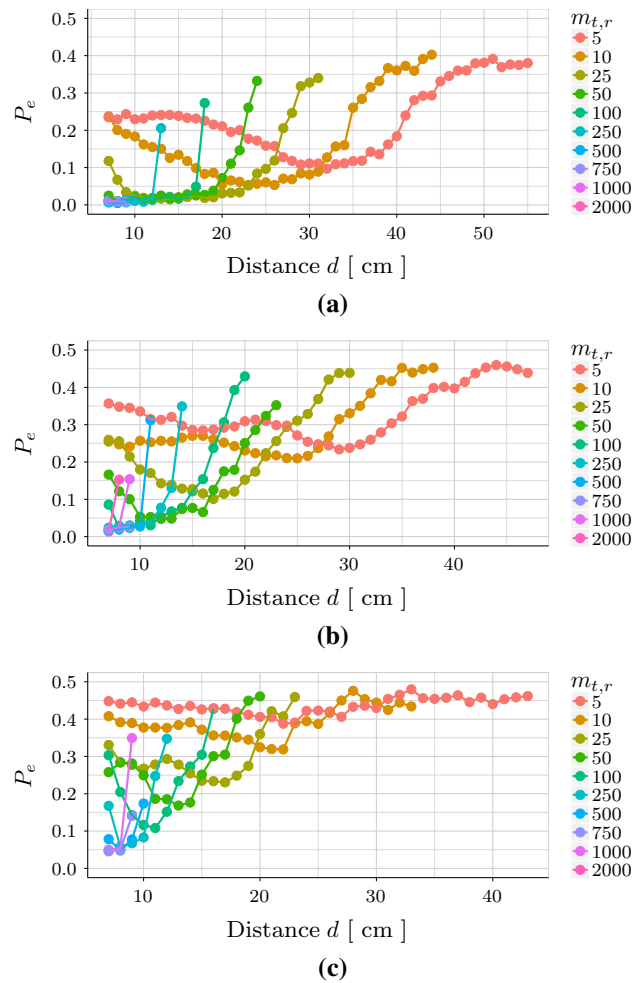


Fig. 10 Bit error probability, P_e , for a detector using a fixed relative decision threshold, $m_{t,r}$, for various distances among two static robots (for details see text). Bit-rates are **a** 310 bps, **b** 650 bps, and **c** 1160 bps

Figure 11 shows the bit error probability for the dynamic detector (using a default relative decision threshold of $m_{t,r} = 5$). The dynamic detector greatly reduces the bit error probability, while retaining a good communication range. For high bit-rate transmissions, high bit error probabilities are observed when the robots are not near each other. This can be avoided by redefining the default relative decision threshold, $m_{t,r}$, thereby limiting the maximum communication range.

Additional experiments were performed to investigate the role of the default relative decision threshold for the dynamic detector in (36). Figure 12a shows how the choice of $m_{t,r}$ impacts on the communication range. Figure 12b provides a lower bound for the relative decision threshold to ensure, given a bit rate, that the bit error probability does not exceed, $P_e < 0.01$.

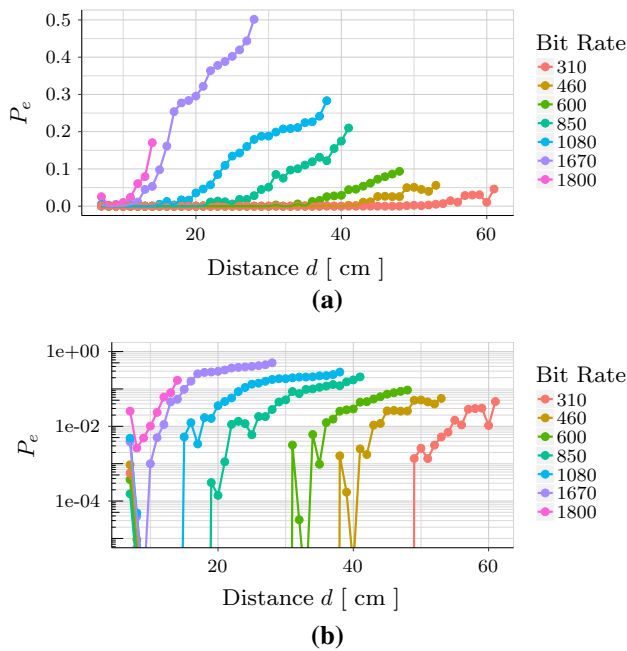


Fig. 11 Bit error probability, P_e , for a detector using a dynamic relative decision threshold for various distances among two static robots. **a** and **b** present the bit error probability linearly and logarithmically, respectively

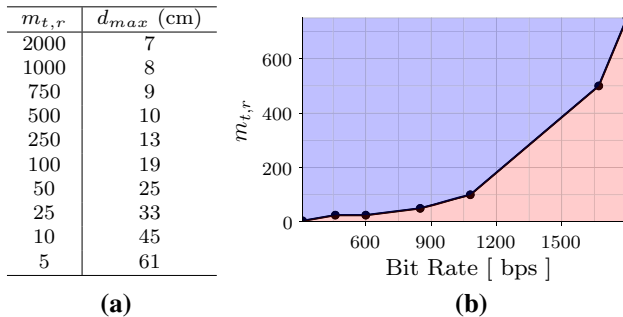


Fig. 12 The dynamic detector can be initialized with different default relative decision thresholds. **a** Shows how the choice of threshold impacts on the communication range for 310 bps transmissions. **b** The blue area (above the curve) shows the default relative decision thresholds for which reliable communication ($P_e < 0.01$) can be achieved at a given bit rate (Color figure online)

5.4 Channel coding

SwarmCom uses per default a [15, 15] BCH code (i.e., no channel coding) to maximize throughput. However, the user can select a [15, 11] BCH code (to correct up to $c = 1$ error) or a [15, 1] BCH code (to correct up to $c = 7$ errors). To investigate the impact of channel coding on the accuracy and throughput, the three channel codes are compared. We consider bit rates of 310 bps and 1800 bps.

As a starting point ($c = 0$), we use the experimentally determined bit error probabilities, P_e , from the experiments with a dynamic decision threshold (see Sect. 5.3). We then

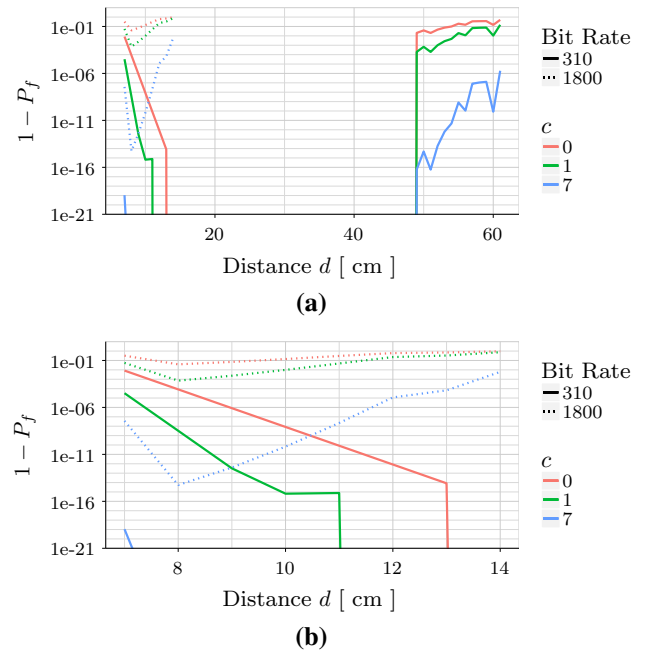


Fig. 13 Two static robots, at relative distance d , use channel coding to correct $c \in \{0, 1, 7\}$ bit errors. Shown is the actual rate (0 bit) and predicted rates (1 bit and 7 bit) at which messages get incorrectly transmitted, $1 - P_f$, at 310 bps and 1800 bps, respectively. Zero values are not visible, due to the log-scale. **a** Full range, **b** zoomed range

compute, for each channel code, the probability of correctly transmitting a 15-bit message, P_f , based on the assumption that errors are independent of each other.

As shown in Fig. 13, the probability of incorrectly transmitted messages, $1 - P_f$, decreases by several orders of magnitude when correcting up to 7 errors (for both bit rates). This should make it possible to transmit messages virtually free of error (i.e., $1 - P_f \leq 10^{-16}$) over a range of up to 50 cm. However, the increase in accuracy would come at the expense of an effectively reduced throughput: $\frac{1800}{15} = 120$ bps and $\frac{310}{15} \approx 21$ bps. Compared with coded 1800 bps transmissions, the uncoded 310 bps transmission is less prone to errors (for $d \geq 11$ cm, that is, ≥ 4 cm gap between the robots), allow a 4-fold communication range, while providing ca. 2.5 times higher throughput. Where throughput is not the primary criterion, however, the application of channel coding is recommended.

5.5 Comparison with libIrcom

To validate SwarmCom, the implementation with a dynamic threshold is compared to the state-of-the-art communication software for the e-puck, libIrcom, which is widely used by the swarm robotics community (Murray et al. 2013; Prieto et al. 2010).

For the libIrcom implementation, identical functions were used to control the robot and exchange messages via Blue-

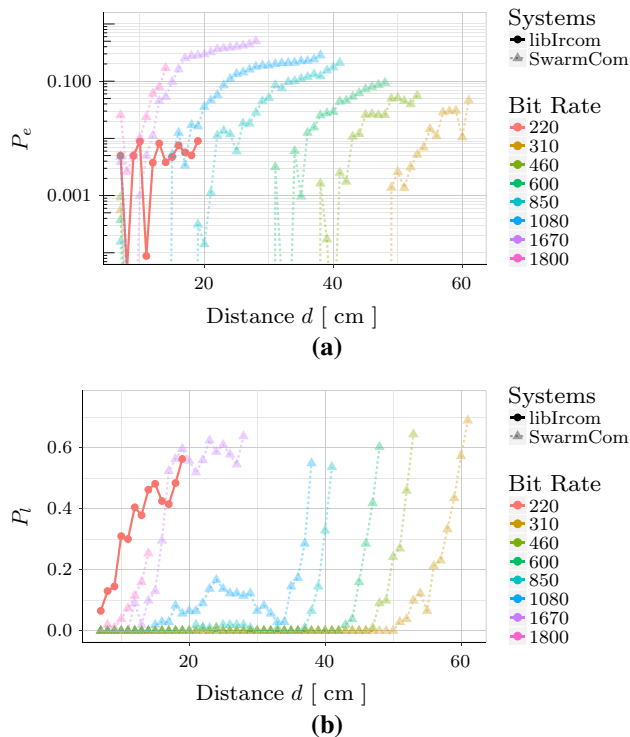


Fig. 14 Comparison of libIrcCom (red) with SwarmCom (other colors): **a** bit error probability, P_e , and **b** probability of message loss, P_l (Color figure online)

tooth with the computer. The experimental setup was the same as described in Sect. 5.1.

Figure 14a shows the bit error probability, P_e , as a function of the inter-robot distance. The bit rate of libIrcCom was experimentally determined as 220 bps. At a bit rate of 330 bps, SwarmCom consistently outperforms libIrcCom over all distances. Moreover, it provides a 3-fold higher communication range and 50% additional throughput. At other bit rates of up to 850 bps, SwarmCom consistently outperforms libIrcCom over all distances as well, while providing up to ca. 380% additional throughput.

Figure 14b shows the probability of message loss, P_l . As can be seen, SwarmCom with bit rates of up to 1080 bps provides a substantially lower probability of message loss when compared to libIrcCom with 220 bps.

We believe that the improved performance of SwarmCom can be attributed to a range of factors. SwarmCom exploits knowledge of the underlying channel model, and offers an adaptive threshold mechanism that allows a robot to adjust its sensitivity to the strength of signals it perceives, thereby increasing the communication range and reducing the error probability. Moreover, it treats every measurement as a logical 0 or 1, thereby maximizing throughput (though, optionally, channel coding can be applied). libIrcCom uses long codewords (and rudimentary channel coding) for the logical 0s and 1s, effectively reducing throughput.

Algorithm 1 Random Walk

Require: t_m , the forward moving interval

```

1: procedure MOVE
2:   loop
3:     Choose a random rotation angle,  $\theta \in (-\pi, \pi)$ 
4:     Rotate by  $\theta$ 
5:     Choose a random velocity,  $v \in [-128, 128] \frac{\text{mm}}{\text{s}}$ 
6:     repeat
7:       Move forward with  $v$ 
8:     until time,  $t_m$ , has elapsed
9:   end loop

```

6 SwarmCom: mobile robots evaluation

To validate SwarmCom in more realistic situations, experiments with groups of mobile robots are conducted. We consider a relatively congested area by choosing a higher robot density than in the static robot evaluation.

The robots communicate with each other while performing a random walk. We consider two modes of communication:

- When a robot, R_t , sends a message via *local broadcast*, the message is received by the neighbors of R_t but does not propagate further through the network. It does not reach robots outside the communication range of R_t . Moreover, due to occlusions, it may not reach a robot unless located in the direct line of sight of R_t .
- When a robot, R_t , sends a message via *flooding*, the message is also locally transmitted at first. However, every robot that has received the message will locally retransmit it too, but only once. Thereby, the message will propagate through the network.

6.1 Experimental setup

The experiments described in this section are conducted in a 72 cm \times 72 cm arena. The arena has a grey floor and is surrounded by 30 cm tall white walls. It is illuminated as described in Sect. 5.1.

By default, we use groups of 2 to 7 e-pucks. Note that we automate and monitor the experiments using Bluetooth, and the Bluetooth protocol does not allow to establish more than seven connections at once.

We follow a similar experimental protocol to the one used in experiments with static robots. The robots use a dynamic threshold [see (36)]. Changes in the level of ambient light are compensated as described in Sect. 4.5. However, the experimental protocol differs in the following points:

- The robots are initially placed in random positions and orientations, chosen arbitrarily from within the arena;
- Each robot performs a simple random walk using Algorithm 1 (with $t_m = 2$ s);

- For every parameter configuration (number of robots, bit rate, communication range), 1000 messages are to be transmitted (previously 100);
- For every message, robot R_t is randomly selected from the group;
- Messages are declared lost after waiting for 2 s (previously 1 s).

The experiments are recorded by a camera mounted on top of the arena. OpenCV (version 3.4) is used to calculate the robots' positions, potential occlusion, and inter-robot distances. This is subsequently used to calculate an estimate of the probability of message loss, \hat{P}_l , in our static analysis.

The procedure to obtain \hat{P}_l depends on the mode of communication. If the local broadcast is used, every robot, R_r , in the direct line of sight of R_t is considered. Once the distance between the pair of robots is established, an estimate for the the probability of message loss is obtained from the relation shown in Fig. 14b. If flooding is used, all non-cyclic paths from R_t to any robot, R_r , are considered. The path of transmission is the one candidate path, p , that provides the largest probability of successful transmission, $1 - \hat{P}_{l,p}$. As a result, P_l is defined by

$$P_l = 1 - \max_p \left\{ \prod_i (1 - \hat{P}_{l,p,i}) \right\}, \quad (37)$$

where $\hat{P}_{l,p,i}$ is the probability of message loss for hop i of path p between R_t and R_r .

To compare the outcome of experiments, a two-sample Kolmogorov–Smirnov test (KS test) is performed (Young 1977). This nonparametric statistical test compares two empirical cumulative distribution functions (CDF)¹¹. Two CDFs, $F_{1,n}$ and $F_{2,m}$, are statistically significantly different when

$$\Delta^{(i,j)} = D_{m,n}^{(i,j)} \sqrt{\frac{mn}{m+n}} \geq \sqrt{-\frac{1}{2} \ln \left(\frac{\alpha}{2} \right)}, \quad (38)$$

where $D_{m,n}^{(i,j)} = \sup_x |F_{i,n}(x) - F_{j,m}(x)|$ is the maximum difference of two CDFs. Here, we choose $\alpha = 0.01$, hence $\Delta^{(i,j)} \geq 1.628$. Note that the obtained distribution of experiment i is tested to produce statistically larger values than another distribution, j , by changing $D_{m,n}^{(i,j)}$ to $G_{m,n}^{(i,j)} = \max_x (F_{i,n}(x) - F_{j,m}(x))$. If not stated otherwise, $D_{m,n}^{(i,j)}$ is used in a KS test (i.e., statistical equality is tested).

In this work, an experiment consists of 6 trials, where each trial has a specific number of robots (i.e., two to seven robots). To compare two experiments, we report KS vectors

¹¹ An empirical cumulative distribution function, $F_{i,k}$, of experiment i is based on a sample of k measurements.

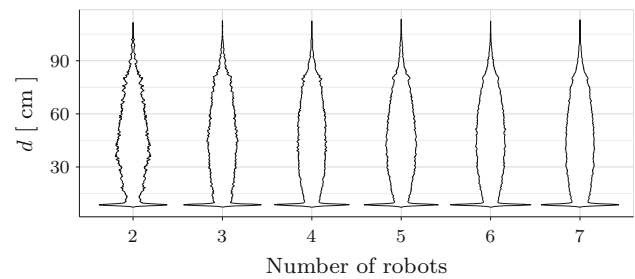


Fig. 15 The distribution of inter-robot distance, d , between two communicating robots for experiments with 2, 3, ..., 7 robots

and matrices. A KS vector contains 6 KS test values, $\Delta^{(i,j)}$. Each value is calculated by comparing the trials with the same number of robots. A KS matrix consists of a $\Delta^{(i,j)}$ for each combination of trials.

6.2 Experiments

This section reports the results of the experiments with multiple mobile robots. First, the distribution of d is analyzed—recall, the robots perform a random walk without obstacle avoidance. Then, the performances of the two modes of communication are compared. The effects of changing either the velocities, communication range, the bit rate, or the density of robots are analyzed. Finally, SwarmCom is compared to libIrcm.

6.2.1 Inter-robot distance

The communication performance of any group of robots will ultimately depend on the spatial distribution. We determine the distribution of inter-robot distances, d , for a single line-of-sight transmission by analyzing the video recordings. Robots whose sight of the transmitting robot is occluded are discarded, as they would be unable to establish a communication channel.

Figure 15 shows the distribution of distances as a violin plot. The distances appear to be distributed similarly over trials for any number of robots. To test whether the distributions are identical, we formulate the null hypothesis that the distributions of d are equal. The resulting KS matrix is as follows:

$$\begin{pmatrix} 0 & 16.941 & 16.973 & 7.310 & 13.336 & 18.702 \\ 16.941 & 0 & 4.281 & 16.420 & 12.791 & 5.135 \\ 16.973 & 4.281 & 0 & 16.249 & 10.632 & 2.883 \\ 7.310 & 16.420 & 16.249 & 0 & 11.587 & 21.165 \\ 13.336 & 12.791 & 10.632 & 11.587 & 0 & 11.831 \\ 18.702 & 5.135 & 2.883 & 21.165 & 11.831 & 0 \end{pmatrix}. \quad (39)$$

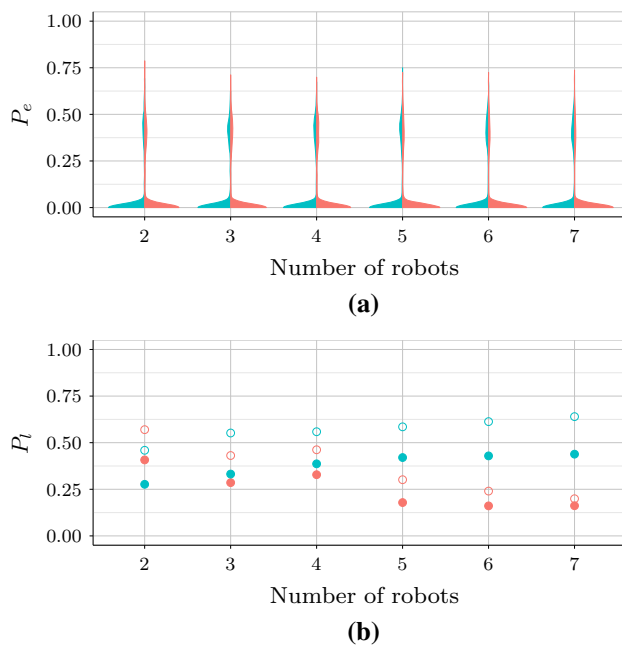


Fig. 16 Experiments by mode of communication: local broadcast (blue) and flooding (red). Shown are **a** the bit error probability, P_e , and **b** probability of message loss, P_l , all grouped by the number of robots. Dots indicate mean values and empty circles indicate \hat{P}_l , which was estimated through computer vision analysis (Color figure online)

As all elements exceed 1.628 (i.e., $\alpha = 0.01$), the null hypothesis is rejected (highlighted in boldface). Consequently, the number of robots significantly changes the distribution of d .

Note that the peak at 7 cm (see Fig. 15) indicates a collision between R_l and R_r . In that situation, the robots' movement is likely to be temporarily restricted. A further restriction is provided by the boundaries of the arena.

6.2.2 Modes of communication

We investigate both modes of communication—local broadcast and flooding. For each mode, an experiment is conducted with 2–7 robots, which transmit with 310 bps over a communication range of up to 45 cm (i.e., $m_{l,r} = 10$).

Figure 16a shows the bit error probability, P_e , obtained from the experiments. The experiments with and without flooding are indicated by red and blue, respectively. The corresponding KS vector,

$$[0.466, 0.336, 1.361, 0.603, 1.452, 1.131], \quad (40)$$

shows that the mode of communication has no significant impact on P_e for $\alpha = 0.01$.

To investigate the effects of the number of robots on P_e , a KS matrix is calculated for each experiment. When comparing the values of the KS matrices,

$$\begin{pmatrix} 0 & 0.727 & 1.504 & 1.545 & 1.465 & \mathbf{2.111} \\ 0.727 & 0 & 1.253 & 1.254 & 1.110 & \mathbf{1.969} \\ 1.504 & 1.253 & 0 & 0.407 & 0.445 & 1.172 \\ 1.545 & 1.254 & 0.403 & 0 & 0.376 & 1.208 \\ 1.465 & 1.110 & 0.445 & 0.376 & 0 & 1.494 \\ \mathbf{2.111} & \mathbf{1.969} & 1.172 & 1.208 & 1.494 & 0 \end{pmatrix}, \quad (41)$$

$$\begin{pmatrix} 0 & 0.513 & 1.204 & 0.592 & 1.307 & \mathbf{1.918} \\ 0.513 & 0 & 1.021 & 1.076 & \mathbf{2.166} & \mathbf{1.775} \\ 1.204 & 1.021 & 0 & 1.299 & 0.320 & 0.930 \\ 0.592 & 1.076 & 1.299 & 0 & 1.426 & 0.976 \\ 1.307 & \mathbf{2.166} & 0.320 & 1.426 & 0 & 0.947 \\ \mathbf{1.918} & \mathbf{1.775} & 0.930 & 0.976 & 0.947 & 0 \end{pmatrix}, \quad (42)$$

for flooding and local broadcast, respectively to (39), it is evident that the bit error probability is less affected by the number of robots than the inter-robot distance. In many cases, P_e does not change significantly with the number of robots. However, when comparing the extremes (2 and 7 robots), P_e significantly differs. As a result, it can be expected that larger groups of robots would increasingly impact P_e .

Note that the distribution of P_e shows two peaks across all experiments (see Fig. 16a). The first peak, at $P_e \approx 0$, is the dominant one. Overall, 68–75% of messages are entirely error-free. The second peak is at $P_e \approx 0.4$, but far less pronounced. A possible cause could be burst errors¹², which could occur when a robot enters or leaves the communication range of another robot during a transmission.

Figure 16b shows the probability of message loss, P_l . Here we extend the definition of (35) to multiple receiving robots, such that $P_l = 0$ implies that the message is received by every robot bar the one that originally transmitted it. As the number of robots increases, the flooding mode outperforms the local broadcast mode, as expected. When using the flooding mode, robots succeed in reaching occluded robots or others outside their local neighborhood. The more robots are present, the more likely it is that the whole group is reached. Note that the more often a message gets retransmitted, the more likely is that some errors get introduced. However, according to (40), no such effect was observed in our experiments.

From Fig. 16b, it can be seen that \hat{P}_l approximates the trend of P_l , though with a noticeable offset. The computer vision analysis only identifies the robot that is sending and then calculates \hat{P}_l based on the topological information over a certain time duration. When robots enter or leave the communication range during a transmission, P_l is not affected, but the computer vision analysis, which decides frame by frame, will be unable to capture this.

¹² Burst errors involve a sequence of successive faulty bits.

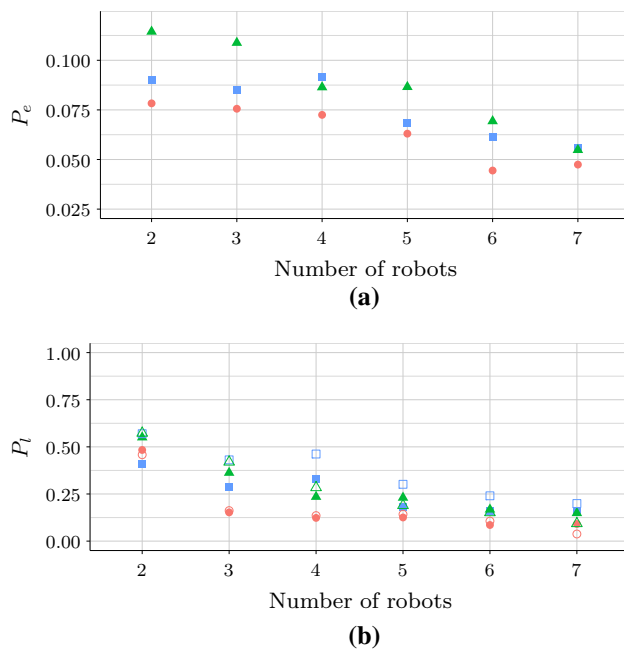


Fig. 17 Experiments with groups of mobile robots moving at $\frac{1}{3}$ (red disk), $\frac{2}{3}$ (blue square) and $\frac{3}{3}$ (green triangle) of their maximum linear and angular velocities. **a** Average bit error probability, P_e ; **b** probability of message loss, P_l . Solid shapes indicate the means of experimentally observed values; hollow shapes indicate the predictions, \hat{P}_l (Color figure online)

6.2.3 Variation of the movement velocities

To investigate how the velocities of robots impact on communication quality, three experiments with 2 to 7 robots are conducted. In these experiments the linear and angular velocities of the robot are limited, respectively, to $\frac{1}{3}$, $\frac{2}{3}$, and $\frac{3}{3}$ of the maximum value. We refer to these three setups as v_1 , v_2 , and v_3 , respectively. The robots perform a random walk (Algorithm 1) and are configured as in the flooding experiment (see Sect. 6.2.2).

Figure 17 illustrates the average values of (a) P_e and (b) P_l . In both cases, the performance appears to improve as the robots move slower.

The three velocity settings (v_1, v_2, v_3) lead to three null hypotheses—that the bit error probability, P_e , for the respective smaller velocity is larger or equal to the bit error probability of the respective larger velocity. A signed Kolmogorov–Smirnov test provides the KS vectors,

$$[1.631, 2.530, 1.902, 2.606, 3.192, 1.325], \quad (43)$$

$$[1.746, 1.721, 2.534, 1.704, 2.306, 1.501], \quad (44)$$

$$[1.715, 1.780, 0.677, 1.970, 1.108, 0.219], \quad (45)$$

for testing settings v_1 against v_2 , v_1 against v_3 , and v_2 against v_3 . As evident from these vectors, the null hypotheses can be rejected (with $\alpha = 0.01$) for trials with few robots. In-

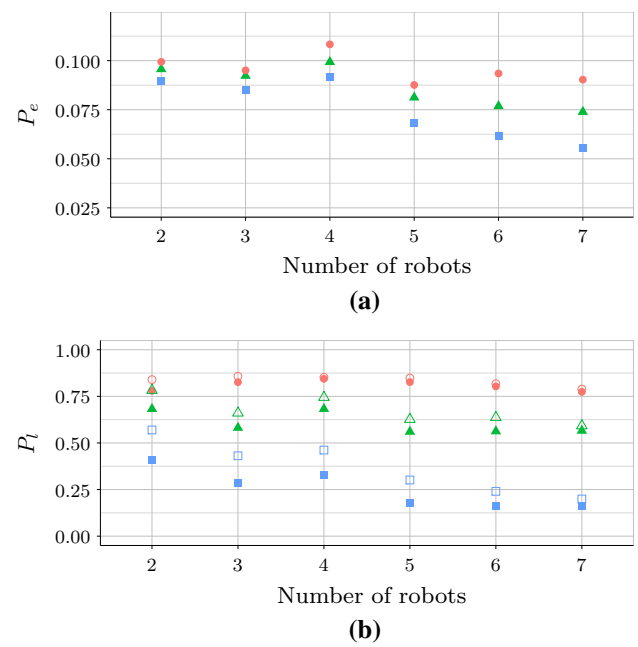


Fig. 18 Experiments with groups of mobile robots using a restricted communication range: **a** the average bit error probability, P_e for communication ranges of 21 cm (red, circle), 33 cm (green, triangle), and 45 cm (blue, square); **b** probability of message loss, P_l . Solid shapes indicate the means of experimentally observed values; hollow shapes indicate the predictions, \hat{P}_l (Color figure online)

creasing the velocity may also increase the likelihood that some robots interfere with others or that a communicating robot prematurely leaves the range of communication during a transmission. However, as the number of robots in the swarm increases, more communication paths are formed, and the effect becomes less prominent.

6.2.4 Variation of the communication range

To investigate the role of the default relative decision threshold (and hence the communication range) in the mobile robot scenario, three experiments were conducted. We use $m_{l,r} = 75, 25$, and 10, which results in a communication range of 21 cm, 33 cm, and 45 cm, respectively. The robots transmit at a bit rate of 310 bps.

Figure 18a illustrates the average values of P_e . To investigate the difference between the distributions of P_e , we consider all three pairs of communication ranges. This leads to three null hypotheses—that the bit error probability, P_e , for the respective smaller communication range is less or equal to the bit error probability of the respective larger communication range. We use a signed Kolmogorov–Smirnov test (Press et al. 2007), which provides KS vectors,

$$[0.607, 1.027, 0.521, 1.075, \mathbf{1.772}, \mathbf{1.799}], \tag{46}$$

$$[1.570, \mathbf{2.266}, \mathbf{2.646}, \mathbf{3.151}, \mathbf{3.610}, \mathbf{4.300}], \text{ and } \tag{47}$$

$$[1.108, 1.440, \mathbf{1.765}, \mathbf{2.494}, \mathbf{4.271}, \mathbf{4.246}], \tag{48}$$

for testing communication range 21 cm against 33 cm, 21 cm against 45 cm, and 33 cm against 45 cm, respectively. As evident from these vectors, unless the number of robots is relatively low, the null hypotheses can be rejected (with $\alpha = 0.01$). In other words, smaller communication ranges produce significantly larger bit error probabilities. We suspect that, as the communication range decreases, it becomes increasingly unlikely for a pair of mobile robots to remain connected for the duration of a transmission. Robots entering or leaving the range while a transmission is in progress will contribute to a higher bit error probability.

Figure 18b plots the probability of message loss. As expected, the shorter the communication range, the higher the losses. Furthermore, \hat{P}_l predicts the trend of P_l consistently, which suggests that P_l mainly dependent on topological information (i.e., communication range).

6.2.5 Variation of bit rate

SwarmCom can operate with various bit rates, offering a trade-off between throughput and communication range. To investigate the effects on P_e and P_l for groups of mobile robots, three experiments with bit rates of 310 bps, 1080 bps, and 1670 bps were conducted. To enable a fair comparison, the communication range was restricted to 21 cm.

Figure 19a illustrates the average values of P_e for each bit rate. To investigate the difference between the distributions of P_e , we consider all three pairs of bit rates. This leads to three null hypotheses—that the bit error probability, P_e , for the respective larger bit rate is less or equal to the bit error probability of the respective smaller bit rate. We use a signed Kolmogorov–Smirnov, which provides KS vectors,

$$[0.609, 0.842, 1.411, 1.057, 1.352, \mathbf{1.779}], \tag{49}$$

$$[0.432, 0.999, 1.013, 1.502, \mathbf{1.876}, \mathbf{2.371}], \text{ and } \tag{50}$$

$$[0.432, 0.613, 0.746, 0.898, 0.705, 1.115], \tag{51}$$

for testing 310 bps against 1080 bps, 310 bps against 1670 bps, and 1080 bps against 1670 bps, respectively. As evident from these vectors, unless the number of robots is relatively large, the null hypotheses cannot be rejected (with $\alpha = 0.01$). In other words, the error probabilities are less influenced by the bit rate as by the communication range (see the previous section).

Finally, Fig. 19b indicates that the bit rate does only have a moderate impact on P_l .

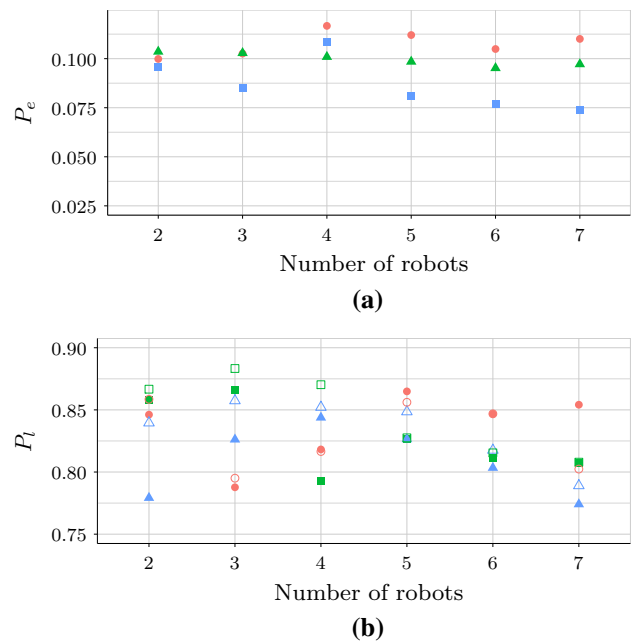


Fig. 19 Experiments with groups of mobile robots using various bit rates: **a** the average bit error probability, P_e for bit rates of 1670 bps (red, circle), 1080 bps (green, triangle), and 310 bps (blue, square), respectively. **b** Probability of message loss, P_l . Solid shapes indicate the means of experimentally observed values; hollow shapes indicate the predictions, \hat{P}_l (Color figure online)

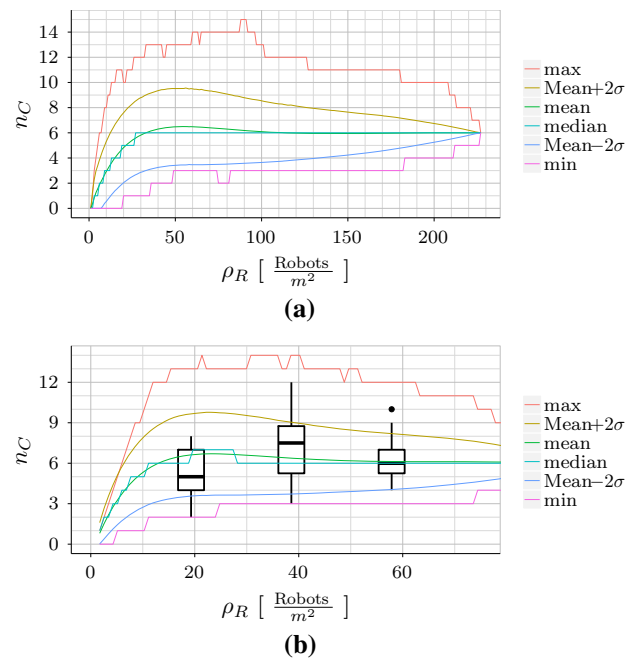


Fig. 20 Number of channels, n_C , a robot can establish depending on the density of robots in its environment. **a** Results from simulations in an open environment. **b** Results from real-robot experiments (shown as box plot) and simulations (shown as line graph) in a bounded environment. For details, see text

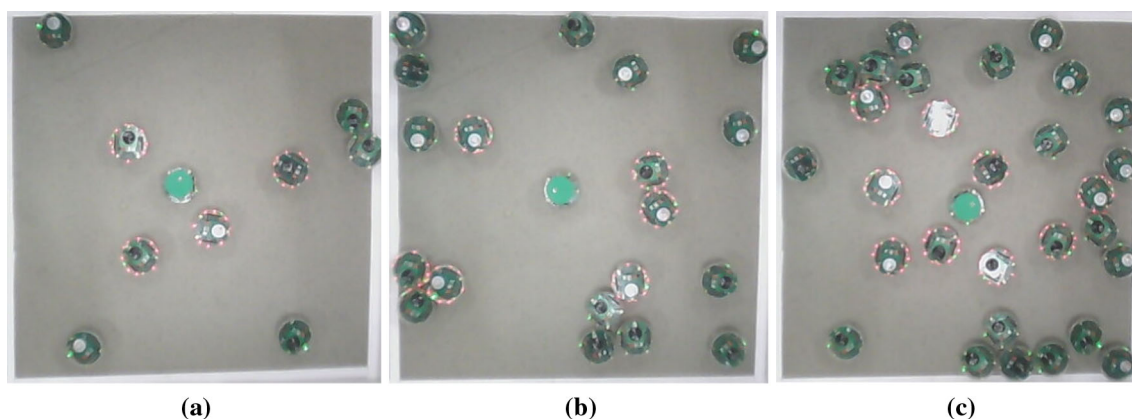


Fig. 21 Snapshots taken from the experiments with **a** 10, **b** 20, and **c** 30 e-puck robots within a 72×72 cm environment. The static robot in the center (green cap) repeatedly transmits a signal. The other

robots, which perform a random walk, activate their LEDs whenever they receive the signal. The number of channels that the central robot establishes are $n_C = 4, 6,$ and $9,$ respectively (Color figure online)

6.2.6 Scalability

To investigate the scaling properties of SwarmCom, two studies were conducted—one in computer simulation, the other using real robots.

The simulation study examines how the robot density impacts on the connectivity of the network. One robot, placed at the center of the environment, transmits messages within a communication range of 61 cm (see Table 12a). The area within the communication range is populated by randomly placing $1, 2, \dots, N$ additional robots, where $N = 272$ was the largest value for which sampling-based positioning proved efficient.¹³ This results in a robot density, $\rho_R \in [1, 234] \frac{\text{robots}}{\text{m}^2}$. All robots are static. For every tested density, 10000 trials are conducted. In every trial, we determine how many channels, n_C , are established by the central robot. We consider a channel to be established if a robot can distinguish between s_0 and s_1 . In other words, the received signal is below threshold $m_{t,dyn}$ (see Sect. 4).

Figure 20a shows the number of connections, n_C , as a function of robot density, ρ_R . It is evident that the number of channels is bounded by a relatively small constant (≈ 15). In other words, while hundreds of other robots may reside within the communication range of the central robot, the latter can only communicate with a relatively small number of robots, all others being occluded. This greatly reduces the load on any node. In fact, the load is bounded by a constant for any number of robots within the swarm. As the robot density increases to its maximum, the number of connections decreases too, until finally settling at $n_C = 6$. In the most dense setting (i.e., corresponding to a near-hexagonal

arrangement of the robots), the central robot will have 6 communication channels.

The real-robot study examines how the robot density impacts on the connectivity in practice. Three experiments are conducted with swarms of 10, 20, and 30 physical robots within a 72×72 cm space as shown in Fig. 21. This results in the robot densities, $\rho_R \in \{19.3, 38.6, 57.9\} \frac{\text{robots}}{\text{m}^2}$. A static robot, placed in the center, transmits a series of s_1 symbols. All other robots perform a random walk (Algorithm 1). When a moving robot detects the signal, it illuminates its LEDs. Every 30 s, the number of illuminated robots, n_C , is determined using video recordings from an over-head camera. The process is repeated 30 times per experiment. In other words, in total 90 experimental trials were conducted.

Figure 20b shows that the real-robot experiments are overall in good agreement with the simulation results.¹⁴ These findings suggest that SwarmCom's resource allocation and utilization is indeed scalable by design, and, discounting delays, could be applied to swarms of any size.

6.2.7 Comparison with libIrcCom

Finally, the overall performance is determined by comparing SwarmCom to libIrcCom in a mobile setup. To allow a fair comparison, SwarmCom is configured to use the same communication range (i.e., 21 cm) and a similar bit rate (i.e., 310 bps).

Figure 22a compares the distributions of P_e of both SwarmCom (red) and libIrcCom (blue). Overall, 68–75% of messages sent with SwarmCom are error-free, which is 3–5 times larger than the corresponding value for libIrcCom (14–20%). libIrcCom produces significantly larger values of P_e

¹³ We expect the theoretical limit to be around 285; see <http://hydra.nat.uni-magdeburg.de/packing/cci/cci285.html>, for a solution to a related, but less constrained, problem (Graham et al. 1998).

¹⁴ To improve comparability, the simulation runs were repeated with the bounded 72×72 cm arena used in the experiments.

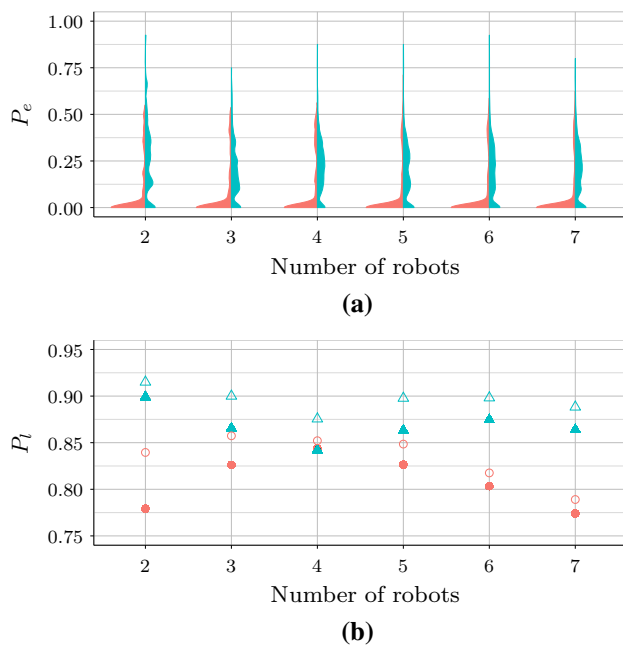


Fig. 22 Experiments with groups of mobile robots comparing SwarmCom (red, circles) and libIrcCom (blue, triangles). **a** Distribution of P_e ; **b** mean of P_l (solid) and \hat{P}_l (hollow) (Color figure online)

than SwarmCom, as evident from the corresponding KS vector,

$$[4.207, 6.397, 7.630, 9.030, 10.316, 12.382]. \quad (52)$$

When comparing (46)–(51) to the KS vector, it can be seen that changing the bit rate or communication range has a less severe impact on the bit error probability than changing to libIrcCom.

Figure 22b plots the probability of message loss for SwarmCom (red, circles) and libIrcCom (blue, triangles). As expected, similar communication ranges result in a comparable probability of message losses.

Similar to previous sections, \hat{P}_l predicts the trend of P_l consistently.

7 Conclusion

This paper proposed the first model to describe the infrared signals transmitted and received on a widely used swarm robotics platform, the e-puck robot. The model can be used by pairs of robots to exchange information. The model was shown to provide a good fit to measurements obtained from experiments that were different from those used during parameter identification.

The second contribution of the paper is SwarmCom, an optical MANET for computationally severely constrained robots and its implementation on the e-puck, a robot with

only 8 kB of RAM. SwarmCom uses a dynamic detector that, by adapting to the environment and nearby robots, achieves orders of magnitudes of improvement in bit error probability. In addition, SwarmCom supports various bit rates and communication ranges to allow users to adapt the system to their needs. Reliable communication is provided up to 61 cm at 310 bps, and up to 14 cm at 1800 bps.

SwarmCom supports channel coding, allowing to correct up to 7 bits per 15-bit block. Based on experimental results without channel coding, our analysis predicts that a pair of robots, employing a 7-bit correction code, can exchange messages over a range of up to 50 cm with a probability of message error of at most 10^{-16} .

To evaluate SwarmCom in a more realistic environment, we conducted experiments with up to seven mobile robots. The flooding communication mode was found to dramatically improve the reachability of other members in the swarm (compared to a local broadcast mode) and did so without significantly affecting the bit error probability. Further experiments showed that shortening the communication range increased the bit error probability as well as limited the network reach.

To investigate how SwarmCom scales with the number of robots, we conducted experiments with up to 30 mobile e-pucks, and computer simulations that increased the robot density until the area corresponding to a robot's communication range was entirely populated with other robots. The results from the experiments and simulations were in good agreement. They showed that the maximum number of communication links per node is relatively low (up to around 15) for any number of robots in the swarm. Consequently, the network's access and utilization operate independently of the total number of robots and, therefore, should be scalable. This results from the line-of-sight transmission mechanism in optical communications: any nodes that are occluded will not be able to establish direct links. In contrast, radio-based signals can reach any node within a communication range. Hence, the maximum robot density must be limited to prevent signal/channel interference. As a result, optical communication provides benefits as offering a reasonable range for miniature robots, and coping well with low-density to high-density situations.

Finally, SwarmCom was compared to libIrcCom—until present, the only infra-red communication software available for the e-puck. We showed that SwarmCom has a 3–5× larger likelihood of error-free transmissions than libIrcCom when using a similar range and bit rate. Overall, SwarmCom can provide 0.8–3 times the range, between 1.4 and 8 times higher bit rates, and between 50 and 63% lower bit error rates than libIrcCom, even when no channel coding is used. This allows the user to prioritize specific parameters—for instance, 3 times larger communication range and 63% lower bit error rates are traded for only 1.4 times of the bit-rate.

In comparison to the static robots experiments, mobile robots experiments showed significant degradation of the communication quality for both SwarmCom and libIrcCom. We showed that higher mobility impacts the system negatively and this should be considered during the design of swarming algorithms. For instance, where throughput is not the primary criterion, channel coding is recommended. To reliably exchange data at high bit rates, robots could temporarily cease motion, or limit their speed, when within communication range, and resume activity thereafter.

A current limitation of SwarmCom—compared to most other MANETs—is its rudimentary routing, which forwards any new message to all possible neighbors. Due to the low throughput, short communication range, and high mobility of the robots, obtaining up-to-date topological information can present a challenge. Flooding therefore remains a valid option. Where the robots engage in frequent, point-to-point communication, however, the flooding protocol is not scalable. Further work on optical communication in swarm robotics (hard- and software) could provide more-reliable and higher-throughput systems as shown in Khan (2017), Zhang et al. (2018). More advanced systems could then extract topological information faster, making it possible to realize more sophisticated routing protocols, as commonly found in MANETs. A further extension could be for SwarmCom to provide situated communication. In other words, in addition to obtaining the content of a message, the receiving robot would get informed about the relative position of the emitting robot. This could prove useful, for example, to maintain the connectivity with certain robots in the local neighborhood.

Our channel model applies not only to the original e-puck platform but also to the e-puck 2, which was released in 2018. The e-puck 2 uses the same proximity sensors and measuring circuit as the original version.¹⁵ It should be noted that SwarmCom can be outperformed, at least in bit rate (up to 5000 bps) and communication range (up to 600 cm), by dedicated communication hardware, such as the e-puck range and bearing board (Gutiérrez et al. 2008, 2009a; Millard et al. 2017). These solutions, however, add additional costs, weight, and energy demand to the default platform.

A question that is open for future work involves the analysis of OpenSwarm in practical applications. The presented analysis relied on the assumption that the robots did not move, or moved at random. A more realistic scenario could involve robots that moved according to the demands of their respective tasks.

Overall, we demonstrated that a rigorous communication model, coupled with a well-designed communication system

could have dramatic effects on swarm communication. Our efforts can be replicated for other communication devices, and our results should encourage further work along these lines.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Anees, S., & Bhatnagar, M. R. (2015). Performance evaluation of decode-and-forward dual-hop asymmetric radio frequency-free space optical communication system. *IET Optoelectronics*, 9(5), 232–240.
- Arvin, F., Samsudin, K., & Ramli, A. R. (2009). A short-range infrared communication for swarm mobile robots. In *2009 International conference on signal processing systems*, (pp. 454–458). IEEE.
- Balch, T., & Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 27–52.
- Bormann, C., Ersue, M., Keranen, A. (2014). Terminology for constrained-node networks (RFC 7228). Internet engineering task force (IETF). <https://tools.ietf.org/html/rfc7228>. Accessed 24 May 2016.
- Caprari, G., Siegwart, R. (2005). Mobile micro-robots ready to use: Alice. In *2005 IEEE/RSJ international conference on intelligent robots and systems*, (pp. 3295–3300).
- Carruthers, J. B., & Kahn, J. M. (1997). Modeling of nondirected wireless infrared channels. *IEEE Transactions on Communications*, 45(10), 1260–1268.
- Chen, C., et al. (2013). Graphene mechanical oscillators with tunable frequency. *Nature Nanotechnology*, 8(12), 923.
- Chen, J., Gauci, M., Li, W., Kolling, A., & Groß, R. (2015). Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, 31(2), 307–321.
- Chlantai, I., Conti, M., & Liu, J. J.-N. (2003). Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 1(1), 13–64.
- Cianci, C. M., Raemy, X., Pugh, J., & Martinoli, A. (2006). Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Simulation of Adaptive Behavior (SAB-2006), Swarm Robotics Workshop* (pp. 103–115). Berlin, Germany: Springer.
- Conti, M., & Giordano, S. (2014). Mobile ad hoc networking: Milestones, challenges, and new research directions. *IEEE Communications Magazine*, 52(1), 85–96.
- Di Caro, G. A., Ducatelle, F., & Gambardella, L. M. (2009). Wireless communications for distributed navigation in robot swarms. In *Workshops on applications of evolutionary computation* (pp. 21–30). Berlin, Germany: Springer.
- Fischer, V., & Hickinbotham, S. (2010). A metabolic subsumption architecture for cooperative control of the e-puck. *Nature inspired cooperative strategies for optimization (NICSO)* (pp. 1–12). Berlin, Germany: Springer.
- Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2010). *Gamma distribution* (pp. 109–113). Hoboken, NJ: Wiley.
- Garattoni, L., & Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Science Robotics*, 3(20), eaat0430.
- Glisic, S. G., & Leppänen, P. A. (2013). *Wireless communications: TDMA versus CDMA*. Berlin: Springer.

¹⁵ Information obtained through mail exchange with GCTronic (<http://www.gctronic.com/>); schematics of the e-puck 2 have not been released at the time of writing this paper.

- Glover, I., & Grant, P. M. (2010). *Digital communications*. New York, NY: Pearson Education.
- Graham, R. L., Lubachevsky, B. D., Nurmela, K. J., & Östergård, P. R. (1998). Dense packings of congruent circles in a circle. *Discrete Mathematics*, *181*(1–3), 139–154.
- Guerroumi, M., Pathan, A.-S. K., Badache, N., & Moussaoui, S. (2014). On the medium access control protocols suitable for wireless sensor networks—A survey. *International Journal of Communication Networks and Information Security*, *6*(2), 89–103.
- Gutiérrez, Á., Campo, A., Dorigo, M., Amor, D., Magdalena, L., & Monasterio-Huelin, F. (2008). An open localization and local communication embodied sensor. *Sensors*, *8*(11), 7545–7563.
- Gutiérrez, Á., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., Magdalena, L. (2009a). Open e-puck range & bearing miniaturized board for local communication in swarm robotics. In *IEEE international conference on robotics and automation (ICRA)*, (pp. 3111–3116). Piscataway, NJ: IEEE.
- Gutiérrez, Á., Tuci, E., & Campo, A. (2009b). Evolution of neuro-controllers for robots' alignment using local communication. *International Journal of Advanced Robotic Systems*, *6*(1), 25–64.
- Haddad, M., Muhlethaler, P., Laouiti, A., Zagrouba, R., & Saidane, L. A. (2015). TDMA-based MAC protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues. *IEEE Communications Surveys Tutorials*, *17*(4), 2461–2492.
- Hauert, S., Leven, S., Zufferey, J.-C., Floreano, D. (2010). Communication-based swarming for flying robots. In *Workshop on network science and systems issues in multi-robot autonomy, IEEE international conference on robotics and automation (ICRA)*. Piscataway, NJ: IEEE.
- Hirschman, K., Tsybeskov, L., Duttagupta, S., & Fauchet, P. (1996). Silicon-based visible light-emitting devices integrated into micro-electronic circuits. *Nature*, *384*(6607), 338.
- Hussain, F., Anpalagan, A., & Vannithamby, R. (2017). Medium access control techniques in M2M communication: survey and critical review. *Transactions on Emerging Telecommunications Technologies*, *28*(1), e2869.
- Khan, L. U. (2017). Visible light communication: Applications, architecture, standardization and research challenges. *Digital Communications and Networks*, *3*(2), 78–88.
- Lapidoth, A. (2009). *A foundation in digital communication*. Cambridge: Cambridge University Press.
- Li, S., Kong, R., & Guo, Y. (2014). Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Transactions on Mechatronics*, *19*(6), 1810–1820.
- Li, Y., Du, S., Kim, Y. (2009). Robot swarm manet cooperation based on mobile agent. In *2009 IEEE international conference on robotics and biomimetics (ROBIO)*, (pp. 1416–1420). Piscataway, NJ: IEEE.
- Macker, J. P., & Corson, M. S. (1998). Mobile ad hoc networking and the IETF. *SIGMOBILE Mobile Computing and Communications Review*, *2*(1), 9–14.
- Malik, A., Singh, P. (2015). Free space optics: Current applications and future challenges. *International Journal of Optics*, *2015*, 945483. <https://doi.org/10.1155/2015/945483>.
- McLurkin, J., et al. (2014). A robot system design for low-cost multi-robot manipulation. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, (pp. 912–918).
- Millard, A. G., et al. (2017). The Pi-puck extension board: A Raspberry Pi interface for the e-puck robot platform. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, (pp. 741–748). Piscataway, NJ: IEEE.
- Mondada, F., et al. (2009). The e-puck, a robot designed for education in engineering. In *9th conference on autonomous robot systems and competitions* (Vol. 1, pp. 59–65). Castelo Branco, Portugal: IPCB.
- Murray, L., Timmis, J., & Tyrrell, A. (2013). Modular self-assembling and self-reconfiguring e-pucks. *Swarm Intelligence*, *7*(2), 83–113.
- Nemec, D., Janota, A., Hrubaš, M., Gregor, M., & Pírník, R. (2017). Mutual acoustic identification in the swarm of e-puck robots.
- Press, W. H., Teukolsky, S. A., Flannery, B. P., Vetterling, W. T. (2007). Kolmogorov–Smirnov Test. In *Numerical recipes in FORTRAN: The art of scientific computing* (pp. 617–620). Cambridge, UK: Cambridge University Press.
- Prieto, A., Becerra, J., Bellas, F., & Duro, R. (2010). Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time. *Robotics and Autonomous Systems*, *58*(12), 1282–1291.
- Rubenstein, M., Ahler, C., Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3293–3298). Piscataway, NJ: IEEE.
- Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, *345*(6198), 795–799.
- Schmickl, T., et al. (2011). CoCoRo—The self-aware underwater swarm. In *IEEE conference on self-adaptive and self-organizing systems workshops (SASOW)* (pp. 120–126). Piscataway, NJ: IEEE.
- Sesay, S., Yang, Z., & He, J. (2004). A survey on mobile ad hoc wireless network. *Information Technology Journal*, *3*(2), 168–175.
- Seyfried, J., Szymanski, M., Bender, N., Estaña, R., Thiel, M., Wörn, H. (2005). The I-SWARM project: Intelligent small world autonomous robots for micro-manipulation. In *Proceedings of the 2004 international conference on swarm robotics* (pp. 70–83). Berlin, Heidelberg: Springer-Verlag.
- Shi, Z., Beard, C., & Mitchell, K. (2013). Analytical models for understanding space, backoff, and flow correlation in CSMA wireless networks. *Wireless Networks*, *19*(3), 393–409.
- Telefunken, V. (1999). *Reflective optical sensor with transistor output*. Vishay Telefunken. <http://zuff.info/doc/TCRT1000.pdf>. Rev. A3.
- Trenkwalder, S. M., Esnaola, I., Lopes, Y. K., Kolling, A., & Groß, R. (2018). Online supplementary material: SwarmCom - an infrared-based mobile ad-hoc network for severely constrained robots. Retrieved July 29, 2019 from <https://naturalrobotics.group.shef.ac.uk/supp/2018-003/>.
- Trenkwalder, S. M., Lopes, Y. K., Kolling, A., Christensen, A. L., Prodan, R., & Groß, R. (2016). OpenSwarm: An event-driven embedded operating system for miniature robots. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4483–4490). Piscataway, NJ: IEEE.
- Tutuko, B., & Nurmaini, S. (2014). Swarm robots communication-base mobile ad-hoc network (MANET). *Proceeding of the Electrical Engineering Computer Science and Informatics*, *1*(1), 134–137.
- Wang, Q., Jaffr-Runser, K., Xu, Y., Scharbarg, J. L., An, Z., & Fraboul, C. (2017). TDMA versus CSMA/CA for wireless multi-hop communications: A stochastic worst-case delay analysis. *IEEE Transactions on Industrial Informatics*, *13*(2), 877–887.
- Young, I. T. (1977). Proof without prejudice: Use of the Kolmogorov–Smirnov test for the analysis of histograms from flow systems and other sources. *Journal of Histochemistry & Cytochemistry*, *25*(7), 935–941.
- Zhang, H., Yang, A., Feng, L., & Guo, P. (2018). Gb/s real-time visible light communication system based on white LEDs using T-Bridge cascaded pre-equalization circuit. *IEEE Photonics Journal*, *10*(2), 1–7.



Stefan M. Trenkwald was born in Innsbruck, Austria. He graduated from the Federal Higher Technical Institute for Educating and Experimenting for Chemistry (HBLVA Rosensteingasse), Vienna, Austria in 2006 and was awarded an Austrian Ingenieur degree in chemistry in 2011. From the Vienna University of Technology (TU Wien), Vienna, Austria, he was awarded a B.Sc. degree in electrical engineering and a BSc degree in computer engineering in 2011 and 2012, respectively.

Since 2013, he is working towards a Ph.D. in swarm robotics at the Natural Robotics Lab at The University of Sheffield. His research resulted, besides others, in the open-source project OpenSwarm, an operating system for swarm robots. His main areas of research interests are embedded operating systems, swarm robotics, distributed systems, and optical mobile communication.



Iñaki Esnaola received the M.S. degree in Electrical Engineering from University of Navarra, Spain in 2006 and a Ph.D. in Electrical Engineering from University of Delaware, Newark, DE in 2011. He joined The University of Sheffield in 2013 as a Lecturer and is since 2018 a Senior Lecturer in the Department of Automatic Control and Systems Engineering of The University of Sheffield. He is also a Visiting Research Collaborator in the Department of Electrical Engineering of Princeton

University, Princeton, NJ since 2013. In 2010–2011 he was a Research Intern with Bell Laboratories, Alcatel-Lucent, Holmdel, NJ, and in 2011–2013 he was a Postdoctoral Research Associate at Princeton University, Princeton, NJ. Since 2018 he serves as the Power System Cyber-security and Privacy Subject Editor for the IET Smart Grid Journal. His research interests include information theory and communication theory with an emphasis on the application to smart grid problems.



Yuri Kaszubowski Lopes received a BSc degree in Computer Science in 2010 and an MEng degree in Electrical Engineering in 2012 from Santa Catarina State University, Brazil and a PhD degree in Automatic Control and Systems Engineering in 2017 from The University of Sheffield. He was research associate in the Department of Electronic and Electrical Engineering at The University of Sheffield from 2016 to 2019. Since 2019 he has been a Lecturer in Analysis of Algorithms

and Computational Complexity at Federal University of Technology - Paraná, Brazil. His research interests are formal methods, distributed computation, swarm robotics, machine learning applied to control, and model predictive control.



Andreas Kolling is a principal scientist and technology lead at iRobot. He received a Ph.D. degree in electrical engineering and computer science in 2009 from the University of California, Merced, and a M.S. degree in computer science in 2006 and B.S. degree in mathematics in 2004 from Jacobs University, Bremen, Germany. From 2013 to 2016 he was an assistant professor in the Department of Automatic Control and Systems Engineering at The University of Sheffield, UK, where he

led the multi-robot systems lab. From 2010 to 2012 he was a post-doctoral research fellow at the Robotics Institute at Carnegie Mellon University and at the University of Pittsburgh. His research interests include planning, multi-robot systems, and human-robot interaction. He has served as a general co-chair for DARS in 2016, as associate editor for ICRA and IROS since 2014, and as guest editor for Autonomous Robots for the special issue 'Distributed Robots: From Fundamentals to Applications'.



Roderich Groß received the Diploma degree (summa cum laude) in computer science from TU Dortmund, Germany, in 2001 and the Ph.D. degree in engineering sciences from Université libre de Bruxelles, Belgium, in 2007. From 2005 to 2009, he was a JSPS Fellow with Tokyo Institute of Technology, Japan; a Research Associate with University of Bristol, UK; a Marie Curie Fellow with Unilever, UK; and a Marie Curie Fellow with Ecole Polytechnique Federale de Lausanne,

Switzerland. Since 2010 he has been with the Department of Automatic Control and Systems Engineering, The University of Sheffield UK, where he is currently a Senior Lecturer. In 2018, he was a Visiting Scientist at CSAIL, Massachusetts Institute of Technology. His research interests include swarm robotics, self-reconfigurable robots, evolutionary robotics, and machine learning. He has served as the General Chair of DARS 2016, Editor of IROS 2015–2019, and as an Associate Editor of Swarm Intelligence, IEEE Robotics and Automation Letters, and IEEE Computational Intelligence Magazine.