Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Reference Models for AutoMoDe

K. Hasselmann, A. Ligot, G. Francesca , and
M. Birattari

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2018-002

February 2018
Last revision: January 2019

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2018-002

Revision history:

TR/IRIDIA/2018-002.001	February 2018
TR/IRIDIA/2018-002.002	April 2018
TR/IRIDIA/2018-002.003	January 2019
TR/IRIDIA/2018-002.004	January 2019

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of IRIDIA. The authors take full responsibility for any copyright breaches that may result from publication of this paper in the IRIDIA – Technical Report Series. IRIDIA is not responsible for any use that might be made of data appearing in this publication.

Reference models for AutoMoDe

Ken Hasselmann, Antoine Ligot,
Gianpiero Francesca, and Mauro Birattari

IRIDIA, Université Libre de Bruxelles, Belgium.

February 2018

1 Introduction

This document describes the different reference models that are used in different versions of AutoMoDe. A reference model is a formalization of the capabilities of a robotic platform used to perform experiments.

AutoMoDe (automatic modular design) is an automatic design approach for robot swarms. AutoMoDe generates, via an optimization algorithm, control software in the form of a probabilistic finite state machine (PFSM). These PFSMs are created by searching for the best combination of preexisting *modules* and the values of their parameters (Francesca et al., 2014).

Each version of AutoMoDe use a reference model in order to apply the method to a specific robotic platform. This document is meant to be upgraded with further adjustment to the reference models for newly conducted experiments around the AutoMoDe method (e.g. Francesca et al. (2015)).

2 Reference model RM1

This reference model is the one used in Francesca et al. (2014) and Francesca et al. (2015). Reference model RM1 is a formalization of the capabilities of the e-puck (Mondada et al., 2009) that are described above: RM1 abstracts sensors and actuators by defining the input and the output variables that are made available to the control software at each control step. Sensors are defined as input variables: the control software can only read them. Actuators are defined as output variables: the control software can only write them. Input and output variables are updated with a period of 100 ms. The reference model RM1 is summarized in Table 1.

According to RM1, the reading of a proximity sensor i is stored in the variable $prox_i$, which ranges between 0 and 1. When sensor i does not perceive any obstacle in a 0.03 m range, $prox_i = 0$; while when sensor i perceives an obstacle closer than 0.01 m, $prox_i = 1$. Similarly, the reading of a light sensor i is stored in the variable $light_i$, which ranges between 0, when no light source is perceived, and 1, when the sensor i saturates. The readings of the three ground sensors are stored in the variables gnd_1, gnd_2 and gnd_3. These variables can take three different values: black, gray and white. The e-puck uses the

Table 1: Reference model RM1.

Input	Value	Description
$prox_{i\in\{1,\dots,8\}}$	[0,1]	reading of proximity sensor i
$light_{i\in\{1,\dots,8\}}$	[0,1]	reading of light sensor i
$gnd_{j\in\{1,2,3\}}$	{black, gray, white}	reading of ground sensor j
n	[0,20]	number of neighboring robots perceived
$r_{m\in\{1,2,\dots,n\}}$	[0, 0.70] m	distance of neighbour m
$\angle b_{m\in\{1,2,\dots,n\}}$	[0, 2π] rad	angle of neighbor m

Output	Value	Description
$v_{k\in\{l,r\}}$	$[-0.12, 0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity

Period of the control cycle: 100 ms

Table 2: Reference model RM1.1: novelties with respect to RM1 are highlighted.

Input	Value	Description
$prox_{i\in\{1,\dots,8\}}$	[0,1]	reading of proximity sensor i
$light_{i\in\{1,\dots,8\}}$	[0,1]	reading of light sensor i
$gnd_{j\in\{1,2,3\}}$	{black, gray, white}	reading of ground sensor j
n	[0,20]	number of neighboring robots perceived
V	$\big([0.5, 20], [0, 2\pi]\big)$	attraction vector

Output	Value	Description
$v_{k\in\{l,r\}}$	$[-0.12, 0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity

Period of the control cycle: 100 ms

range-and-bearing board to perceive other e-pucks in its neighborhood. The variable n stores the number of the neighboring e-pucks. For each neighboring e-puck $m_{j\in\{1,2,3\}}$, the variables r_m and $\angle b_m$ indicate the range and the bearing, respectively. The wheel actuators are operated by the control software through the variables v_l and v_r, in which the control software writes the target linear velocity for the left and right wheel, respectively. The linear wheel velocity ranges between -0.12 m/s and 0.12 m/s.

3 Reference model RM1.1

Reference model RM1.1 is a slightly modified version of RM1, This reference model is summarized in Table 2. In this section we describe the differences between RM1 and RM1.1, any undiscussed matter is assumed to be the same as in RM1 (see section 2 for details).

The only difference with RM1 resides in the detection of neighboring robots. In this reference model the robots do not have access to the angle and distance to all the neighboring robots, but only to an attraction vector of all neighboring

Table 3: Reference model RM2: novelties with respect to RM1.1 are highlighted.

Input	Value	Description
$prox_{i\in\{1,\ldots,8\}}$	[0,1]	reading of proximity sensor i
$light_{i\in\{1,\ldots,8\}}$	[0,1]	reading of light sensor i
$gnd_{j\in\{1,2,3\}}$	{black, gray, white}	reading of ground sensor j
n	[0,20]	number of neighboring robots perceived
V	$\big([0.5,20],[0,2\pi]\big)$	attraction vector
b	[0,20]	number of messaging neighbours perceived
V_b	$\big([0.5,20],[0,2\pi]\big)$	attraction vector

Output	Value	Description
$v_{k\in\{l,r\}}$	$[-0.12, 0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity
s	$\{on,\ off\}$	broadcast state

Period of the control cycle: 100 ms

detected robots. This vector is computed as follows:

$$V_b = \begin{cases} \sum_{m=1}^{n}(\frac{1}{1+r_m}, \angle b_m), & \text{if robots are perceived;} \\ (1, \angle 0), & \text{otherwise.} \end{cases}$$

where n, r_m, and $\angle b_m$ are defined in Table 1

4 Reference model RM2

Reference model RM2 is a modified version of RM1.1, This reference model summarized in Table 3. In this section we describe the differences between RM1.1 and RM2, any undiscussed matter is assumed to be the same as in RM1.1 (see section 3 for details).

The variables highlighted are the elements of novelty with respect to RM1.1: b, V_b, and s.

In RM2, every robot locally broadcasts the message by setting a specific bit of its heartbeat's payload. Due to this extension, at every time step, a robot can infer the number and relative position of the neighbouring peers that are broadcasting the message. The information that is made available to the control software is stored in the variables b and V_b. The former is the number of neighbouring peers that broadcast the message and the latter is a an attraction vector to those peers. Formally,

$$V_b = \begin{cases} \sum_{m=1}^{b}(\frac{1}{1+r_m}, \angle b_m), & \text{if } b > 0 \text{ broadcasting robots are perceived;} \\ (1, \angle 0), & \text{otherwise.} \end{cases}$$

Here, r_m and $\angle b_m$ are the range and bearing of the m-th neighbouring peer that is broadcasting the message. The variable s can be set by the control software and indicates whether, during the following control cycle, the robot should broadcast the message or not. It can take two values: *on* or *off*.

Table 4: Reference model RM2.1: novelties with respect to RM2 are highlighted.

Input	Value	Description
$prox_{i\in\{1,\dots,8\}}$	[0,1]	reading of proximity sensor i
$light_{i\in\{1,\dots,8\}}$	[0,1]	reading of light sensor i
$gnd_{j\in\{1,2,3\}}$	{black, gray, white}	reading of ground sensor j
n	[0,20]	number of neighboring robots perceived
V	$([0.5, 20], [0, 2\pi])$	attraction vector
$b_{i\in\{1,\dots,2^\ell\}}$	[0,20]	number of messaging neighbours perceived
V_{b_i}	$([0.5, 20], [0, 2\pi])$	attraction vector

Output	Value	Description
$v_{k\in\{l,r\}}$	$[-0.12, 0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity
$s_{i\in\{1,\dots,2^\ell\}}$	$\{on,\ off\}$	broadcast state

Period of the control cycle: 100 ms

5 Reference model RM2.1

Reference model RM2.1 is a modified version of RM2, This reference model summarized in Table 4. In this section we describe the differences between RM2.1 and RM2, any undiscussed matter is assumed to be the same as in RM2 (see section 4 for details).

The variables highlighted are the elements of novelty with respect to RM1.1: b_i, V_{b_i}, and s_i.

In RM2.1, every robot locally broadcasts a message i by setting a specific bit of its heartbeat's payload. Due to this extension, at every time step, a robot can infer the number and relative position of the neighbouring peers that are broadcasting message i.

The value of i (the message being sent or received) can take a number of values depending on the number of selected message size ℓ (in number of bits). The information that is made available to the control software is stored in the variables b_i and V_{b_i}. The former is the number of neighbouring peers that broadcast the message and the latter is a an attraction vector to those peers. Formally,

$$V_{b_i} = \begin{cases} \sum_{m=1}^{b} i\left(\frac{1}{1+r_m}, \angle b_m\right), & \text{if } b_i > 0 \text{ broadcasting robots are perceived;} \\ (1, \angle 0), & \text{otherwise.} \end{cases}$$

Here, r_m and $\angle b_m$ are the range and bearing of the m-th neighbouring peer that is broadcasting the message.

The variable s_i can be set by the control software and indicates whether, during the following control cycle, the robot should broadcast the message i or not. It can take two values: *on* or *off*.

References

Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., Reina, A., Soleymani, T., Salvaro, M., Pinciroli, C., Mascia, F., Trianni, V., and Birattari, M. (2015). AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2/3):125–152.

Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65, Castelo Branco, Portugal. IPCB: Instituto Politécnico de Castelo Branco.