

Simulation of Hardware Variations in Swarm Robots

Beining Shang, Richard Crowder, Klaus-Peter Zauner
Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton
Southampton, SO17 1BJ
Email: {bs3g10, rmc, kpz}@ecs.soton.ac.uk

Abstract—Swarm robotic systems can offer advantages of robustness, flexibility and scalability, just like social insects. One assumption that is made by the majority of swarm robotic researchers, particularly in software simulation, is that a robotic swarm is a group of identical robots, there is no difference between any two of them. However, differences among hardware robots are unavoidable. These hardware differences, albeit small, affect the robots response to its environment. In this work, robots with hardware variation have been modeled and simulated in a line following scenario. It is found that even small hardware variations can result in behavioral heterogeneity. Although the variations can be compensated by the controllers in training, the hardware variation and resulting differences in controller settings are amplified in the non-linear interaction between robot and environment. Accordingly, the behavior of the identically trained robots in the same environment are subject to divergence.

Keywords—Mobile Robots; Robot Control; Swarm Robotics

I. INTRODUCTION

Swarm robotic research is either simulation-based or hardware-based. When simulating robotic swarms it is quite common to make the assumption that each robot is identical in all features.

It has been argued in [1] that swarm robots are heterogeneous at hardware level, even if hardware-based robots are manufactured to the same design. In addition the authors further argued that although it is very difficult and not cost-effective, identical behaviors of hardware-based robots in a swarm is still possible, in which software is fine-tuned to compensate the inherent hardware differences [1]. Pugh and Martinoli's [2] simulated obstacle avoidance experiments showed that genetic algorithms and particle swarm optimization can withstand different type of sensor variations (offsets and scaling factors). Most researchers have focused on utilizing different software procedures or adopting different hardware equipments to develop homogeneous swarm systems [3], [4]. However it can not be ignored that hardware variations always exist when building a robotic swarm. Such variations can be found in sensor sensitivity, motor characteristics, etc. We have demonstrated through simulation that even though such hardware variations may be very small, they can still largely influence robots' behaviors because of the amplifying impact of the robots' controllers and environment.

In this paper, we consider the performance characteristics of a number of simple robots suitable for operation in swarm, that contain variations in their component parameters. Even

after optimization, they are still subject to widely varying performance. In Section II we discuss the sources of hardware variations. In order to investigate the effect of component variation in a simple scenario, we simulated a line following robot. The approach we followed is outlined in Section III, with the details of the robot used in Section IV. Prior to placing the robots into the test arenas, the controller parameters were optimized through a training process detailed in Section V. Following training, the simulated robots undertook path following tasks within two separate arenas (structured and unstructured), the results of these experiments are discussed in Section VI. The paper concludes with implications for the design of swarm algorithms in Section VII.

II. PROBLEM DESCRIPTION AND ASSUMPTION

A. Hardware Variation

In a hardware-based swarm, the physical robots are built either by hand or using mass production techniques. Although they all appear the same and are often regarded as homogeneous robots, they are not truly homogeneous. Some hardware-related variations always exists [5], due to the factors summarized in Fig. 1.

Differences between any two seemingly identical robots can emerge both when they are manufactured and then when used. Variations result from differences in the manufacture procedures including components specification and assembly [6]. For example, the same type of sensors used on any two robots can have marginally different sensitivity [7]. In addition actuators can have different output profiles and batteries have different characteristics under load. Once the robots have been assembled, the positions of components and soldering parameters vary. For instance, the orientation of IR sensors [8] or wheels will be subject to variation. Finally when in use, different robots experience different circumstances of wear and tear or even damage, such as sensor aging, decreasing battery capacity, and mechanical deterioration. This is especially the case in swarms where the material and calibration cost per unit has to be very low.

B. Assumption

Of all these differences, variations of sensors and actuators are most obvious in terms of influence on robot behavior (cf., Fig. 2). Variations in robotic sensors can cause a robot to perceive different information which is then sent to the controller. The controller, a highly non-linear system, outputs

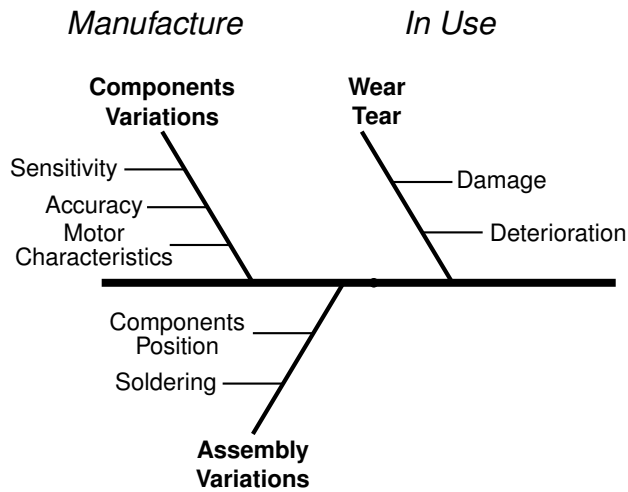
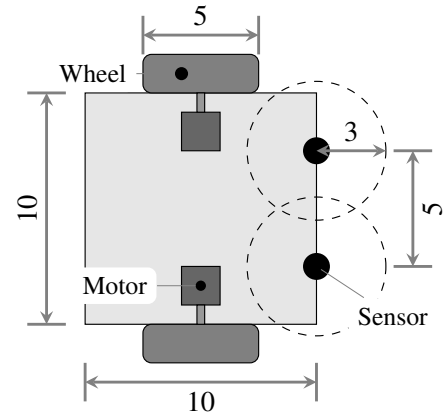


Fig. 1. Overview of possible reasons for hardware difference within swarm robots built to an identical design.



(a) Plan view of the robot, showing the two motors for the differential steering and the two IR sensors. Dimensions are in centimeters.

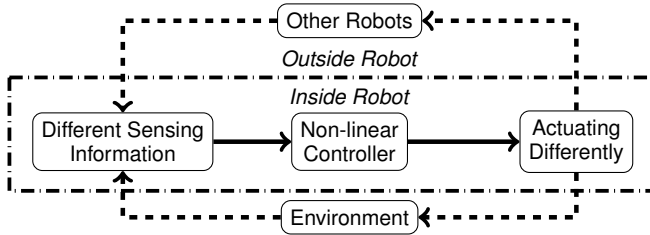


Fig. 2. Sensors and Actuators Variations Influence Robotic Behavior

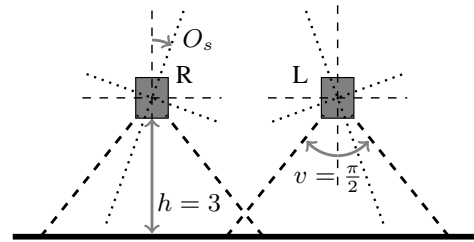
different commands to the actuators. As a consequence of the actuation the sensory input changes leading to another cycle through the interaction loop between robot and environment.

III. METHODOLOGY

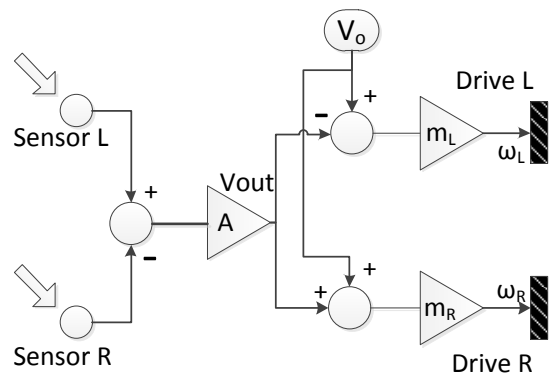
To test this assumption, a small number of robots with different hardware parameters were modeled and their behaviors in a number of line following scenarios compared. The simulation-based approach was selected for this work because it is easy to accurately and consistently create robots with hardware variations, than actually build individual robots. Following training over a fixed path, to allow the individual robot's control parameters to be determined, the robots were used in two controlled experiments. At this stage of the work we are investigating the impact on performance, with respect to a single hardware change per robot. A line following scenario was chosen because robotic behaviors (i.e trajectories) can be easily compared.

IV. SYSTEM MODELING

To accurately model a practical system it was assumed that the robot follows a highly reflective track or line, requiring the robot being fitted with two IR photoelectric sensors. The basic features of the robot being simulated are shown in Fig. 3, the robot is based on a conventional, differentially steered, two wheeled robot fitted with casters to the front and rear. Details of the robot's model are discussed in the following sections.



(b) Front view of the sensor arrangement. The viewing angle is v , while misalignment in assembly is represented by the individual offset angle O_s and changes in the height h . In addition the sensors can be individually angled toward either the front or rear of the robot, O_f .



(c) The control arrangements for the robot. The difference between the two sensors is amplified using an analog PI amplifier, the resultant output is used to modify the individual motor speed's.

Fig. 3. The robotic base used in the simulation.

A. IR Sensor

Two downward pointing IR sensors are located at the front of the robot, Fig. 3(b). The sensor's response to an individual line element is modeled using (1),

$$S(x, \theta) = \beta + \frac{\alpha}{x^2} \cos(\theta) \quad \text{Within } \angle v \quad (1)$$

where the sensor's output is a function of the incidence angle of the reflective light θ and the distance (x) between the sensor and the reflecting point, while α determines the sensitivity of the sensor, and β models the sensor's output offset and the effect of ambient light. The parameter $\angle v$ is IR sensor's viewing angle.

As the line being followed can be considered to be multiple consecutive points, reflecting light to the sensor, the magnitude of IR sensor output can be obtained by summing the result of (1) for each dot within the sensor's viewing angle, $\angle v$. Therefore the output from each sensor can be considered to be represented by (2). The controller is represented by the following equation in the simulator;

$$S_T = \sum_1^n S(x_n, \theta_n) \quad \text{Within } \angle v \quad (2)$$

where n is the number of dots within each sensors viewing angle.

B. Controller

In a line following scenario, the robot's controller will try to keep the output of left and right IR sensors identical, if not the robot will change its relative position to the line to correct any for in-balance. The controller being modeled is shown in Fig. 3(c), where the difference of the sensor values is fed to a PI (proportional and integral) amplifier and used to offset the input voltage to the two motor drives.

$$V_{out} = p \delta_t + i(\delta_t + \delta_{t-1} + \delta_{t-2}) \quad (3)$$

where δ_t is difference between the outputs of the two sensors at time step t , and p and i are the proportional and integral gains.

C. Driving Train

A conventional differential steering driving train was used, where the two driven wheels are powered by brushed d.c. motors, as shown in Fig. 3(a). The d.c. motors are modeled as pure gain, m between the output of the controller and the wheel speed (this simplifies the modeling of the d.c. brushed motor and any gearbox). In the control scheme used, the voltage, V_o , determines the forward speed.

$$\begin{aligned} \omega_L &= (V_o + V_{out}) m_L \\ \omega_R &= (V_o - V_{out}) m_R \end{aligned} \quad (4)$$

Based on the speeds of the individual motors, the effective linear velocity of the wheels can determined using the radius

of the individual wheels r_L and r_R , hence the robot linear (\dot{x} and \dot{y}) and turning $\dot{\phi}$ speeds can be calculated:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\frac{r_L \sin \phi}{2} & -\frac{r_R \sin \phi}{2} \\ \frac{r_L \cos \phi}{2} & \frac{r_R \cos \phi}{2} \\ -\frac{r_L}{b} & \frac{r_R}{b} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \quad (5)$$

where b is the distance between the wheels.

D. Variant design of a swarm robot

As discussed in Section II, the objective of this work is to investigate the differences caused by variations resulting from individual component differences or assemble variations. Within the current model the following variations can be accommodated:

- *Component Variations:* IR sensor sensitivity (α), IR sensor viewing angle ($\angle v$), motor gearbox gain (m) and the wheel radius (r).
- *Assembly Variations:* IR sensor height (h), IR sensor lateral offset (O_s), IR sensor forward offset angle (O_f).

In order to simulate the hardware variations, a group of robots with individual variations were created as summarized in Table. I. In practice, hardware variations typically follow a Gaussian distribution, however in this work, in order to clarify how hardware variation influence swarm robotic behaviors, fixed variations were used. The values of the robot's parameters are all reduce by 1%, except IR sensor lateral offset parameter (O_s) and IR sensor forward offset angle (O_f), which were offset by 5° , away for the centre axis of the robot.

TABLE I. INDIVIDUAL PARAMETERS FOR EACH OF THE SIMULATED ROBOTS. IN EACH OF THE FOURTEEN ROBOTS ONE PARAMETER WAS CHANGED. THE FIFTEENTH ROBOT WAS CONSIDERED TO BE IDEAL, WITH NO VARIATIONS.

Left			Right	
No.	Variation	Components	Variation	No.
L1	-1%	α	-1%	R1
L2	-1%	$\angle v$	-1%	R2
L3	-1%	h	-1%	R3
L4	-5°	O_s	-5°	R4
L5	-5°	O_f	-5°	R5
L6	-1%	m	-1%	R5
L7	-1%	r	-1%	R7

V. TRAINING

Once defined, all the fifteen robots were individually trained in order to optimize the coefficients of the PI controller. A sine wave shaped line was used in the training arena, as shown in Fig. 4. In training all the robots start from the starting point on the left of the arena and are oriented to the right in order to reach the end point on the right of the arena, Fig. 4.

A simple controller parameters optimization scheme was used. For each robot, the controller parameters are selected so that minimum positional and orientational errors were achieved over the training runs. The optimization process was separated into two stages.

Initially the proportional gain, p , was varied, while the integral gain, i , was held at zero. After each trial, the position

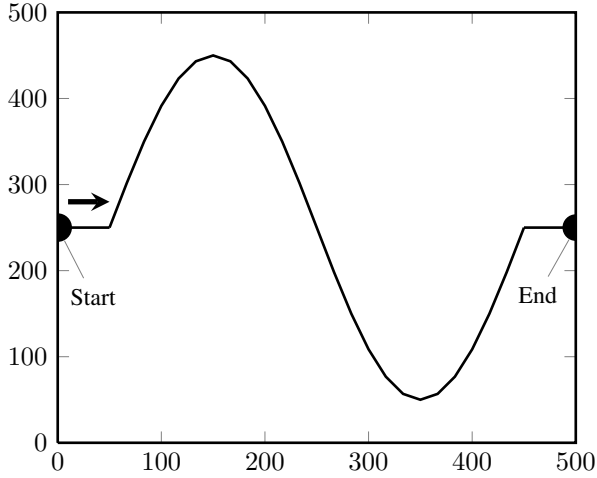


Fig. 4. The training arena, which has a dimension of 500×500 arbitrary units. The line to be followed was a sinusoid with two straight elements,

error of specific robot's trajectory was averaged. The proportional gain at which achieved the minimum average position error, was selected. The position error of robot at each time step was obtained by evaluated the minimum distance between a robot's position and the line to be followed.

In the second stage, the integral gain, i , was varied while the proportional gain was held constant at $1.2 \times p$. After each trial, the orientation error of specific robot's trajectory is averaged. Orientation error of robot at each time step can be obtained by evaluated the absolute difference between robot's orientation and the tangent of nearest point on the line. The coefficient i which allowed the robot to achieve the minimum average orientation error was selected.

After training, while the robots have different controller parameters due to different hardware variations, they all achieved excellent performance over the training arena. The average position error and orientation error of all trained robots have been plotted in Fig. 5, each dot represents one robot and they converge in a very small area, with an average position error $\in [0.16, 0.28]$, and average orientation error $\in [0.019, 0.023]$.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

After training, to test their behaviors, all the robots were required to follow lines in both structured and nonstructured environments as shown in Fig. 6 and Fig. 7. In each testing scenario, the robots start from the identical point shown in the figures, with exactly same orientation. As noted in Section III, the robots are tested individually and not as a swarm, so the robots were not required to interact with anything, except the lines.

Fig. 6 and Fig. 7 show the structured and unstructured environments used to compare the robots to the line-following environment. The thick lines in the arena are the lines that robots are required follow and the thin lines are robots' trajectories. In the Fig. 6, the lines the robots are required to follow are symmetrical about $y = 250$. In the second test environment the lines are randomly located across the arena.

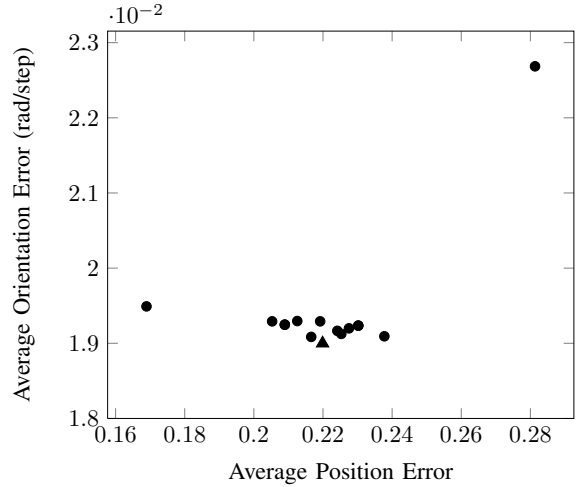


Fig. 5. The positional errors in the robots following training. The ideal robot is identified by a triangle, with outliers are robots R4 to the left and L4 to the upper left. The position error is in centimeters and the angular error is in radians.

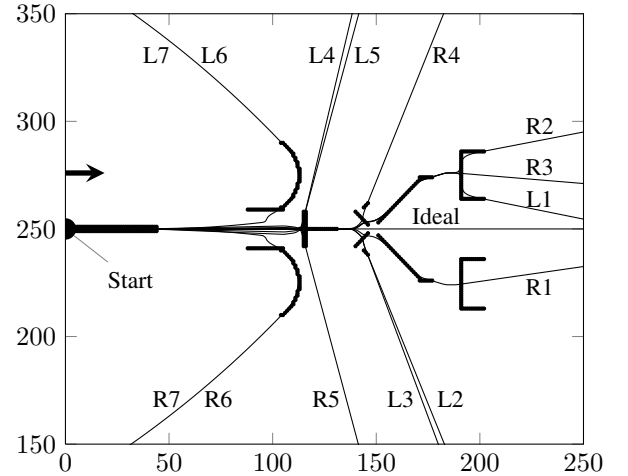


Fig. 6. The testing scenario using a structured, symmetrical environment: The thick black lines are the lines that robots have to follow, it should be noted that the lines are not continuous but contain a number of gaps. The thin lines are trajectories of individual robots as identified in Table I. The ideal robot with no variation always goes through the environment following a linear trajectory.

A. Discussion

The simulated robots used in this set of experiments are all variants of an ideal robot, with a single parameter being changed by a fixed amount, as shown in Table 1. Even though these difference are very small, it was shown that robots still behave very differently in terms of their trajectories in both testing environments even when the hardware variations have been compensated by the trained controller.

1) *Variation of Driving Train and Wheel Radius:* In the first testing environment, the trajectories of robot L6 (left motor's gain reduced by 1%) and robot L7 (radius of left wheel reduced by 1%) are identical. This is because these magnitude variations have an identical impact on robots' trajectories. A similar situation occurs to robot R6 and R7. Robots with variations on the left driving train (L6 and L7) and robots

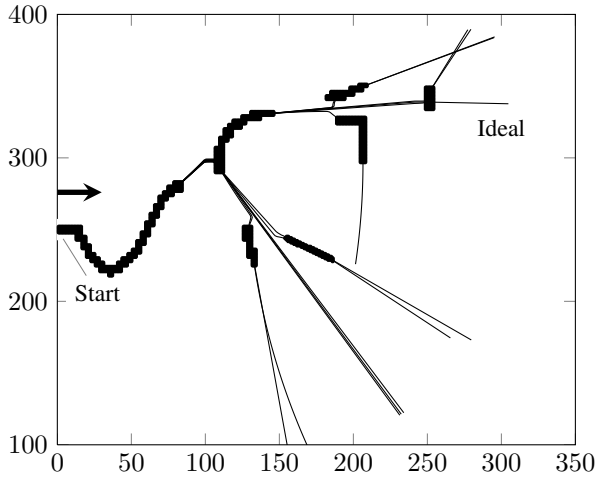


Fig. 7. The testing scenario using the unstructured environment: The thick black lines are the lines that robots have to follow. The thin lines are trajectories of robots. The trajectory of the ideal robot is identified.

with variations on the right driving train (R6 and R7) initially diverge and then follow an arc-shape trajectories.

2) *Variation of Sensor Orientation*: As shown in Fig. 5, using the training procedure presented in Section V, the results show that a larger positional and orientation error is present compared to robots with other types of variations. It can be noted that this type of hardware variations present difficulties for the controller and the training procedure to fully compensate for. Since the variation is only added to one side of the robot, it enables the robot to locate the line in regions not normally possible by the other sensor. It can be noted from Fig. 6, that robots L4, L5, and R5 with variations in their sensor orientation diverge at the identical point in the environment.

3) *Variation of Sensor Viewing Angle*: Robots with variation in their sensor viewing angle, as well as in the height of sensor position h also diverge from the ideal trajectory. The left sensor of robot L3 is nearer to the ground which makes the robot perceive a narrower area, equivalent to the reduction of the viewing angle, as present in robot L2. In the experiment, robots L2 and L3 both went right when encountering the V-shape funnel feature at $(x = 150, y = 250)$ in the environment. This is because the values of their left sensor output is less than that of the right sensor when they are in the middle of the funnel structure. In this case, their controllers compensate this inequality and force robots go right (downwards in Fig. 6), with the robots diverging. Comparing with robot's R2 and R3 separation at the point $(x = 190, y = 250)$, variation in the viewing angle can be considered to have more influence than the height of sensor position.

4) *Variation of Sensor Sensitive*: Robots with variation in the sensor sensitivity also diverge. As seen in Fig. 6, Robot R1 and L1 separate at $(x = 150, y = 250)$, this is also because as their controllers try to compensate for the reduction in sensitivity, the robots follow different routes.

5) *The Unstructured Environment*: The second testing scenario shows an environment consisting of a number of lines randomly drawn to represent the unstructured environment

which robots will typically encounter in reality, again the lines are not continuous, Fig 7. It is clearly shown that robots trajectories are very different, and these difference are all resultant from the small variations to the robot parameters added at the beginning of the simulation.

VII. CONCLUSION

In this work, fifteen robots, including one ideal robots and sixteen robots derived from the ideal one by adding minor variations in either individual components or assembly, have been simulated. This group of robots were first trained individually to optimize the coefficients of their *PI* controllers. In the training task the robots have to follow a continuous line. As a result of the training small hardware variations are compensated by the controller and in general the performance of all robots in the training task is practically the same. When the trained robots are subsequently simulated in an arena where they are exposed to intermittent lines, their paths diverge. This shows that in the model described small hardware variations percolate up to macroscopic distinct behavior. The robots which are all nearly identical and all trained in the same way cannot be assumed to be a homogeneous group. Available algorithms for swarm robots often assume identical robots and focus on the emergent global behavior of such homogeneous swarms. While long the domain of simulations, swarms of hundreds of robots are now becoming also feasible in hardware. Real swarms will necessarily show hardware variations. It will therefore be necessary to contemplate swarm algorithms in the context of diverse robot populations. We view the model system explored here as a testbed for exploring the effect of small unavoidable variations in hardware on the performance of algorithms for swarm robots.

ACKNOWLEDGEMENT

This work was supported by *Panzhuhua ChangNing Co., Ltd.* The authors also acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

REFERENCES

- [1] T. Elliott and N. R. Shadbolt, "Developmental robotics: Manifesto and application." *Philosophical Transactions of the Royal Society of London, Series A*, vol. 361, pp. 2187–2206, 2003.
- [2] J. Pugh and A. Martinoli, "Parallel Learning in Heterogeneous Multi-Robot Swarms," in *IEEE Congress on Evolutionary Computation*, Singapore, September 2007, pp. 3839–3846.
- [3] M. Dorigo, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. Christensen, A. Decugnière, G. Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Régnaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. Turgut, and F. Vaussard, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," Université Libre de Bruxelles, Tech. Rep. TR/IRIDIA/2011-014, 2011.
- [4] L. Parker, "Heterogeneous multi-robot cooperation," Ph.D. dissertation, MIT, February 1994.
- [5] Z. Roth, B. Mooring, and B. Ravani, "An Overview of Robot Calibration," *IEEE Journal of Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.

- [6] G. Alici and G. Shirinzadeh, "A Systematic Technique to Estimate Positioning Errors for Robot Accuracy Improvement Using Laser Interferometry Based Sensing," *Mechanism and Machine Theory*, vol. 40, pp. 879–906, 2006.
- [7] J. Lobo and J. Dias, "Relative Pose Calibration Between Visual and Inertial Sensors," *International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.
- [8] L. Korba, S. Elgazzar, and T. Welch, "Active Infrared Sensors for Mobile Robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, pp. 283–287, 1994.