2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Multi-robot control and tracking framework for bio-hybrid systems with closed-loop interaction

Frank Bonnet[1], Leo Cazenille[2,3], Alexey Gribovskiy[1], José Halloy[2] and Francesco Mondada[1]

***Abstract*— Bio-mimetic robots can interact with groups of animals in bio-hybrid systems to study their behaviour by producing calibrated stimuli and by analysing their responses. Integrating a group of robots into a group of animals to mimic their behaviour is challenging, both in terms of robotic hardware design and robot control. In particular, the robots must be able to react in real-time to the animal changes of behaviour. This implies the need to adequately track and identify animal behaviour.**

In this paper, we present a novel framework to control several bio-mimetic robots to integrate into groups of fish. Our framework is able to track the position of the fish and robots in real-time. The robots are driven by a bio-mimetic model of fish behaviour from the literature. We show that our multi-robot system can successfully integrate into groups of fish with closed-loop interactions between the robots and the fish.

I. Introduction

One of the long-standing interests in ethology is to understand mechanisms of communication and relationships within an animal group. Originally rather simple mock-ups were used by researchers to study these mechanisms, and, naturally, these devices could not support complex interactions or demonstrate any level of adaptation. However nowadays, with the technology becoming more advanced and affordable, these simple instruments are being replaced by robotic devices, capable of interacting with the animal and adapting their behaviour to it. A number of recent works in ethology have successfully used robots to investigate individual and collective animal behaviours, in particular by creating bio-hybrid systems: robots are mixed with cockroaches in [1], [2], chicks in [3], fruit flies in [4], honeybees in [5], [6], guppies in [7] and zebrafish in [8], [9], [10], [11].

The analysis of existing approaches in the field of bio-hybrid systems [12], [13], [14] shows that generally researchers develop the necessary software to perform a closed-loop control of the robots in bio-hybrid systems by themselves, as no such specific software is available commercially. This in-house software is usually not distributed, the only exception being the MADTraC library [15] developed at Princeton University, Princeton, NJ, and that was used in [12] to track a group of golden shiners (*Notemigonus crysoleucas*) and a model of a three-spined sticklebacks (*Gasterosteus aculeatus L.*) moved by a MiaBot Pro wheeled robot. MADTraC provides a desired functionality, but unfortunately its support has been discontinued for a long time with the last changes going back to 2011. An option would be using ROS (Robotic Operating System) [16] that has a distributed and modular design, and implements its own navigation stack and bindings with the OpenCV computer vision library [17], but the current version doesn't have support for multi-robot systems [18], also it is still rather robotic research oriented and thus potentially demanding a significant learning effort; we were on the contrary willing to build a tool easy to use by researchers in the field of behavioural biology. Also ROS is only fully supported on the Unix-like systems, which is rather limiting, as we target biologists who are often Windows or MacOS users.

In this paper we present a novel framework for experimentation on bio-hybrid systems consisting of fish and robots; this framework includes a robotic fish system, an experimental setup and the CATS (Control and Tracking Software) software serving to track animals and robots, and to control the robots. This work is the continuation of the earlier results presented in [19], [20], [21]; the first novelty here is the series of adaptations applied to the robots, the setup and the software to perform a robust tracking of robots and fish, and the second is that the robotic control implements a two-level bio-mimetic animal behavioural model that selects the motion target based on the other fish positions in the same way as they do it, and that commands robot's locomotion pattern to correspond to one of the fish. Also we give a thorough introduction to the CATS software and on the robots' firmware that was not addressed in such detail in the earlier publications. The result of our work is the currently most advanced framework for bio-hybrid systems with a unique combination of the following characteristics: multi-robot closed-loop control and long duration experimentation.

II. Materials and Methods

A. Ethics statement

The experiments performed in this study were conducted under the authorisation N°2778 delivered by the Department of Consumer and Veterinary of the Canton de Vaud (Switzerland) and the Buffon Ethical Committee (registered to the French National Ethical Committee for Animal Experiments #40) after submission to the French and Swiss state ethical board for animal experiments.

[1]F. Bonnet, A. Gribovskiy and F. Mondada are with the Robotic Systems Laboratory, School of Engineering, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland frank.bonnet@epfl.ch

[2]L. Cazenille and J. Halloy are with Université Paris Diderot VII, Sorbonne Paris Cité, LIED, UMR 8235, F-75205 Paris, France leo.cazenille@gmail.com

[3]L. Cazenille is with Sorbonne Universités, UPMC Univ Paris 06, CNRS, ISIR, F-75005 Paris, France

B. Animals and housing

The fish that we selected as a model to conduct this study is the zebrafish *Danio Rerio*. The fish were 18 months old at the time of the experiments. We kept the fish under laboratory conditions, 27 °C, 500 S salinity with a 10:14 day:night cycle. The fish were reared in 55 liters tanks and fed two times a day (Special Diets Services SDS-400 Scientific Fish Food). The water pH level was maintained at 7.5, and Nitrites (NO2) were below 0.3 mg/l.

C. FishBot and RiBot

The FishBot robot was first introduced in [19]. The hardware design of the FishBot is presented in Fig. 1 (right). The FishBot is a wheeled mobile robot with differential drive configuration. The power supply is done continuously through electric cables (brushes) that slip against two conductive plates situated under the aquarium floor and on the support on which the robot moves. This allows long duration experiments with groups of animal and robots. A Bluetooth antenna is used for wireless communication to control the FishBots. Six infrared (IR) proximity sensors are used to detect obstacles and avoid collisions. The new version of FishBot presented in this paper also provides two coloured LEDs underneath the robot. Every robots is assigned an unique colour that is later used by the tracking system to track and to identify this robot from below, it will be discussed in more details in the next sections.

The modular event-based architecture for mobile robots library (ASEBA) has been used to individually control the FishBots in real-time and re-program them during an experiment without flashing the microcontrollers firmware. The control of FishBots motion is done through events that are sent from CATS and that contain the parameters for the locomotion (Fig. 3). Most of the complex behaviours of FishBot are implemented in CATS as described in the next section, and the robots receive the resulted motor commands thanks to the ASEBA event-based protocol. However the collision avoidance behaviour is fully implemented on-board of each robot. The reason for this is that we are using arenas that can be composed of corridors or small rooms, and thus a fast and reliable obstacle avoidance behaviour is required to avoid the collision of the robots with the boarders of the arena or other robots, that otherwise could generate a failure of the experiment with a FishBot blocked or a decoupling between a RiBot and a FishBot. We implement a local obstacle avoidance mechanism based on the IR proximity sensors of the FishBot. Once an obstacle is detected, a corresponding event is sent over the ASEBA network to inform CATS, and the collision avoidance routine is executed: the robot will avoid the obstacle autonomously using a Braitenberg based obstacle avoidance algorithm [22]. Other event emitted by FishBot is "power-down" sent in case of the powering issues, in this case the control software can modify the behaviour of the robot to overcome this situation; also the user is informed on the issue via the user interface.

The RiBot design was first introduced in [20] and [21], and is presented in Fig. 1 (left). This device was designed based on a 3D scanned zebrafish in order to mimic the external shape of the animal. The RiBot is 1.5 times longer than the average length of our zebrafish, due to the size of the selected components. It is equipped with a stepper motor to reproduce the tail beating movements of zebrafish. The RiBot is not able to swim autonomously underwater, and therefore it is linked by magnetic coupling to a FishBot moving below the experimental tank, which allow it to reproduce the fish typical motion. For the control of the RiBots tail beating movements, a Raspberry PI, on which LIRC library is run, is connected to the same network as the main computer, and RC5 signals are generated on an output pin connected to an IR emitter. The IR signal is broadcast over the whole aquarium and received by all the RiBots.

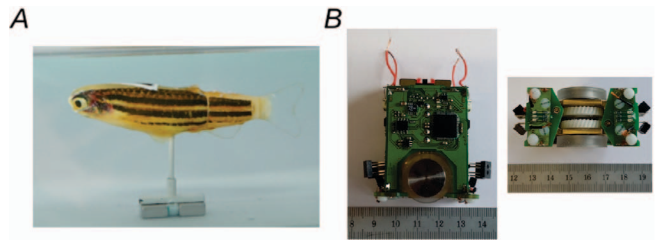

Fig. 1. Panel A: RiBot lure used during the experiments. Panel B: Side and below view of the FishBot wheeld mobile robot used for mimicking fish motion patterns. Compared with previous published version, LEDs are placed below the robot so that it can be tracked using a camera from below.

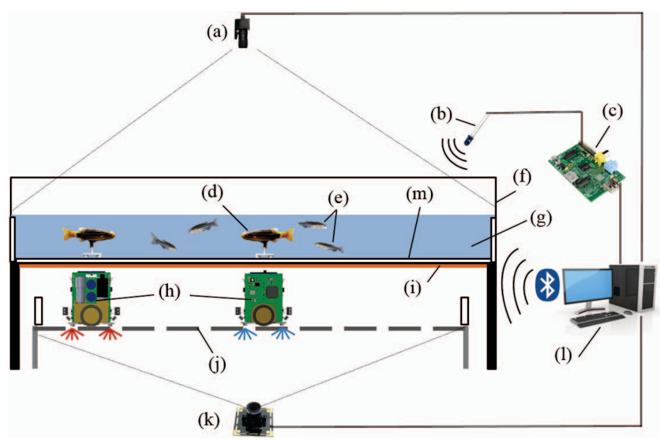

Fig. 2. The design of the experimental setup. (a) Basler camera used grab high definition frames to track the RiBot robotic lure and the zebrafish. (b) IR emitter to control the RiBot. (c) Raspberry PI. (d) Fish-lure RiBot inside the aquarium linked to the mobile robot through magnetic coupling. (e) Zebrafish. (f) Aquarium of 1000 × 1000 × 300 mm. (g) Water layer of 60 mm depth. (h) FishBot mobile robot moving under the aquarium. (i) Copper conductive plates to power the mobile robot (VCC). (j) Perforated stainless steel plates to serve as ground contact for the FishBot (GND) and to observe the FishBot LEDs from below. (k) 180 degrees Fisheye camera to track the FishBot from below. (l) The control station that runs CATS tracking and control software.

D. Experimental setup

The experimental setup consists of an arena made of white plexiglass (Fig. 2) placed in a $1000 \times 1000 \times 300$ mm experimental tank. The size and shape of the arena can vary depending on the experiment. The tank is filled with water up to a level of 60 mm. The floor of the tank is covered with a sheet of polytetrafluoroethylene to provide a smooth surface for the motion of the RiBot robotic fish-lure (described detail in the next section). The whole setup is exposed to diffused light and confined behind white sheets to isolate experiments and homogenise luminosity.

The FishBot mobile robots are powered by two conductive plates, one glued onto the bottom of the aquarium and one onto the a plexiglass support on which the robot moves. The latter is made of stainless steel perforated with 1 mm diameter holes. This modification allowed us to track the LEDs of FishBots with a 180 degree Fisheye camera (elpcctv, China) installed under the setup while still being able to continuously power the robots.

As in our previous studies, an overhead camera acA2040-25gm monochrome GigE CCD camera (Basler AG, Germany) with a maximum resolution of 2048×2048 pixels and equipped with low distortion lenses CF12.5HA-1 (Fujinon, Tokyo, Japan) grabs frames from above the setup that are then processed to track robots and fish.

The experimental PC runs the Control And Tracking Software (CATS) that performs a closed-loop control of the robots by tracking fish and robots and by triggering the corresponding response in the robots.

E. CATS control and tracking software

The design of the software was defined by the requirements. In our case we are building a software for the experimentation of the bio-hybrid multi-agent fish-robot school, where robots interact with animals, and thus a robust and real-time tracking system that is closely tightened with the robot control system was required. The software will be used in different types of experiments and thus both the tracking and control parts must be easily expandable to implement desired behaviour or functionality. Also, the tracking part might be used separately when a pure animal experimentation is run. Hence, the design must be modular to be able easily extract the tracking functionality. The last but not the least, the software is mainly used by the biologists, and thus the user interface must be clean and clear; also the software must be multi-platform.

The overview of CATS software is presented on Fig. 3. It consists of two main components: *Tracking* that tracks the agents, and *Robot control* that generates commands to control the robots. Both tracking and control run at a frequency of 15 Hz, which is enough to make the robots reproduced most of the zebrafish movements. CATS is implemented in C++ with extensive use of the Qt framework [23]. We use the library aravis [24] to access the main Basler camera. All video stream operations are handled using the GStreamer library [25]. We tuned the parameters of the GStreamer media components to have a very low latency. Inside CATS the video frames transfer between different modules is done via single-producer, single-consumer lock-free queues developed in [26].

The tracking routines are based on the OpenCV 3 library [17]. The tracking of the FishBots is performed on the low-resolution (500×500 pixels) video stream coming from the main camera (Fig. 4). The use of low-resolution frames instead of high-resolution ones allow the tracking process to be less computationally expensive. First, we apply a background subtraction preprocessing step, on each frame, by using the Gaussian Mixture-based Background/Foreground Segmentation method described by [27]. The position of the agents is detected by using a corner detection method on the resulting foreground frame: the head of the fish and lures has a very sharp corner. We use the Shi-Tomasi method [28]. In parallel the video stream from the camera below the setup (640×480 pixels) is processed to track individual robots. The robots' positions and orientations are computed based on the detection of coloured blobs in the HSV colour space. First, a threshold is applied to the image, and then the morphological opening operation is applied to the resulted binary image that is followed by the morphological closing [29]. Finally, we detect contours in the resulted image that correspond to FishBots' LEDs [30]. Since every FishBot has its own unique colour there is practically no risk of confusion.

All positions are converted from pixels to real-world coordinates using the calibration routine based on the Tsai's calibration technique [31], moreover the calibration data of both cameras are synchronised and thus we can merge these data. As a result we can efficiently tag the robots among all agents detected in the main camera stream (Fig. 6).

The architecture of the tracking sub-system is modular: other kinds of tracking methods or robot control policies can be added easily, and several parts of the software can be reused in other projects.

The control subsystem implements different types of control modes and locomotion patterns that the user can either select on a configuration file, specify in a so called "control map" or adapt during an ongoing experiment. The control mode defines a behaviour that generates either a target position or a target speed. The target speed is generated by a "random-walk" control mode when a FishBot goes straight forward, while avoiding obstacle; or by a "manual" control mode where the robot is controlled by the user with a joystick. Examples of a control mode generating a target position would be "trajectory-following" or "fish-model". The former orders the robot to follow a set of predefined waypoints, and the latter, recently added to CATS, makes the robot to mimic a fish behaviour in a school; the particular fish model that we have implemented is described in more detail in the next section.

When a target position is provided it is the "locomotion pattern" that defines how the robot goes to this position. We provide two locomotion patterns: the constant linear speed pattern and the fish-like pattern. When applying the constant linear speed locomotion pattern a FishBot follows the target at a parametrised constant speed. The control is done using

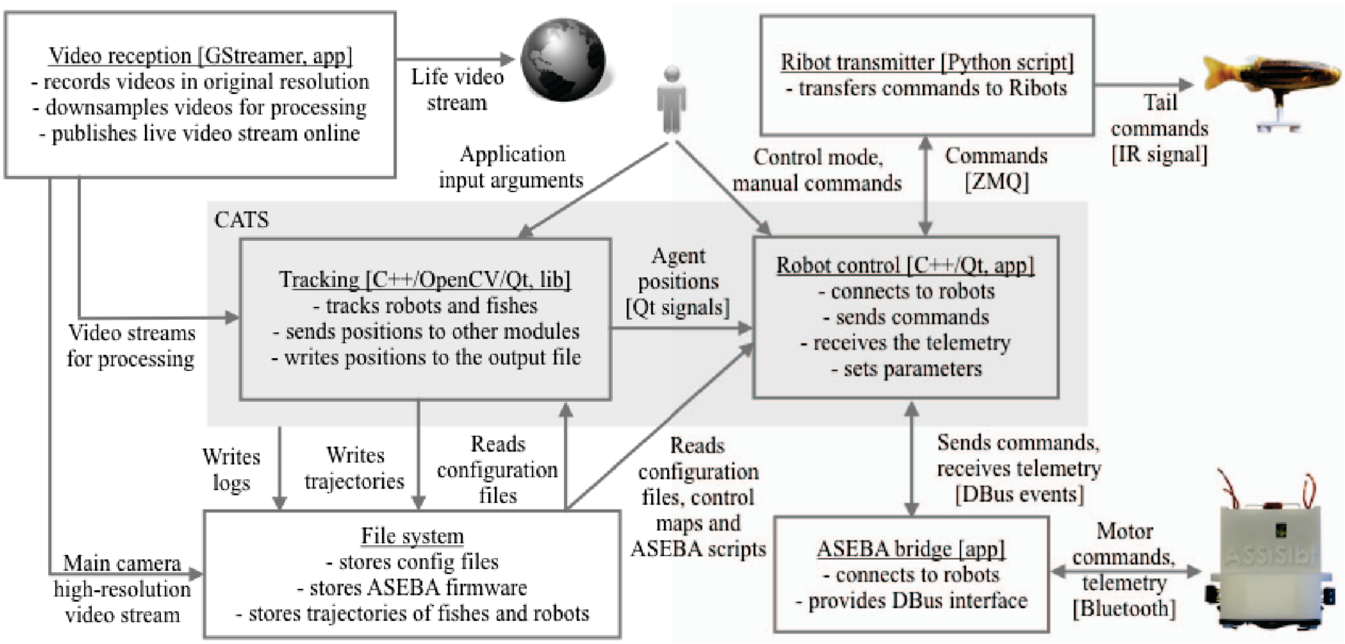

Fig. 3. Overview of the Control And Tracking Software (CATS), used to save videos of the experiments, to track in real-time the positions of the fish and of the robots, and to control robot behaviour. The video stream from the main camera fixed above the setup is compressed and saved on disk in high resolution (2040 × 2040 pixels) for further analysis. It is also converted to a lower resolution (500 × 500 pixels) and published on the Internet (streamyfish.com). The tracking of the fish and RiBots is performed in real-time on the low-resolution video stream. The second video stream (640 × 480 pixels) from the camera fixed under the setup is used to track individual FishBots. The robot control makes use of the tracked positions of the robots and fish to control the FishBots motion as well as RiBots body movements. Low-level control of the FishBot mobile robots is achieved by using the ASEBA framework.

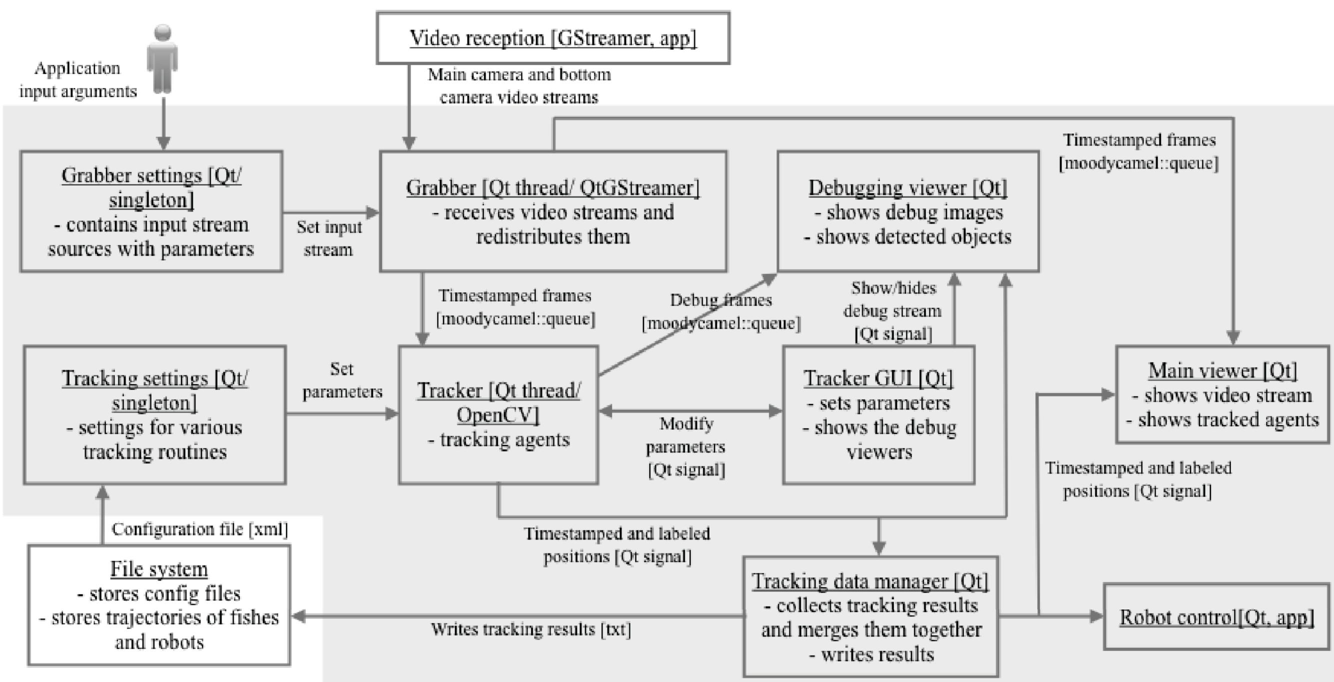

Fig. 4. Overview of the tracking sub-system of CATS. The stream coming from the main camera is used to track all the agents on the arena, while the bottom camera's stream is processed to detect the robots' positions. The tracking results are later merged together to separate fish from the RiBots. The resulted positions are stored in the file system for further analysis, but are also sent to the robot control sub-system. Several tracking methods are available, which methods to use is defined in the configuration file. The software provides a GUI that displays input video streams with the tracking information.

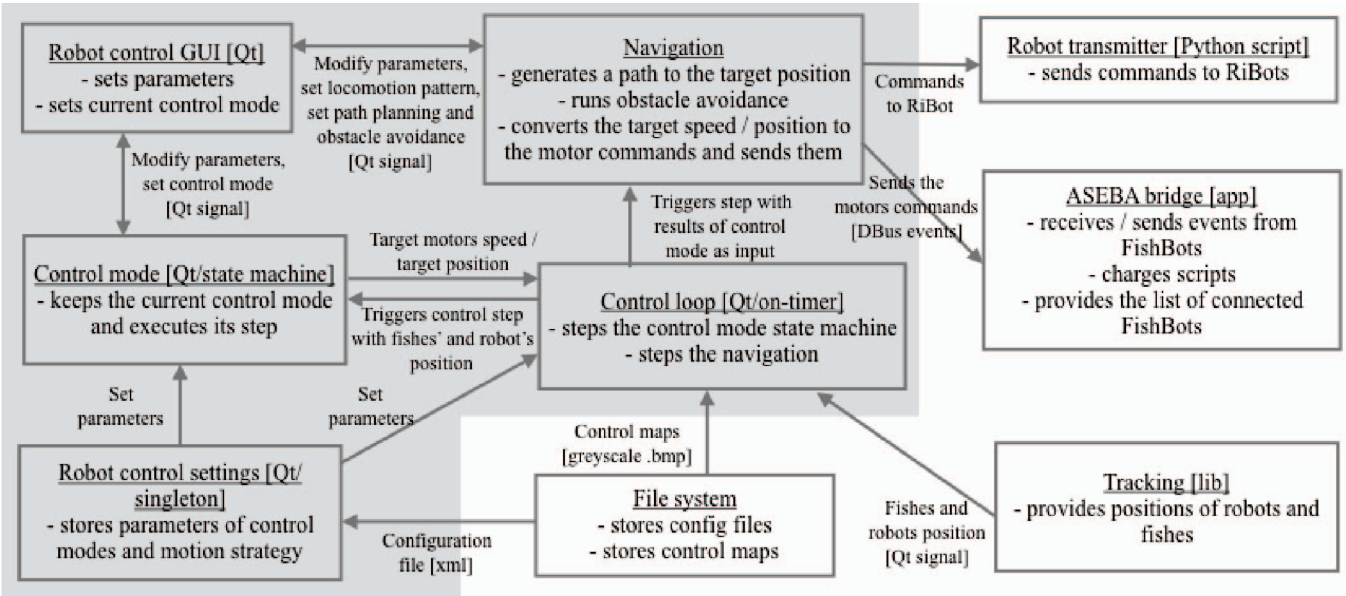

Fig. 5. Overview of the robot control sub-system of CATS. Several kind of control modes (behaviours) and locomotion patterns are available, and can be selected by the user in the user interface. For the FishBots, the desired speed of each wheel is sent through a serial connection via the ASEBA network, in which all the FishBots are connected and can receive or emit events. For the RiBots, the communication is only one way, where the IR RC5 signal is broadcast to control stepper motors that drive the caudal peduncles.

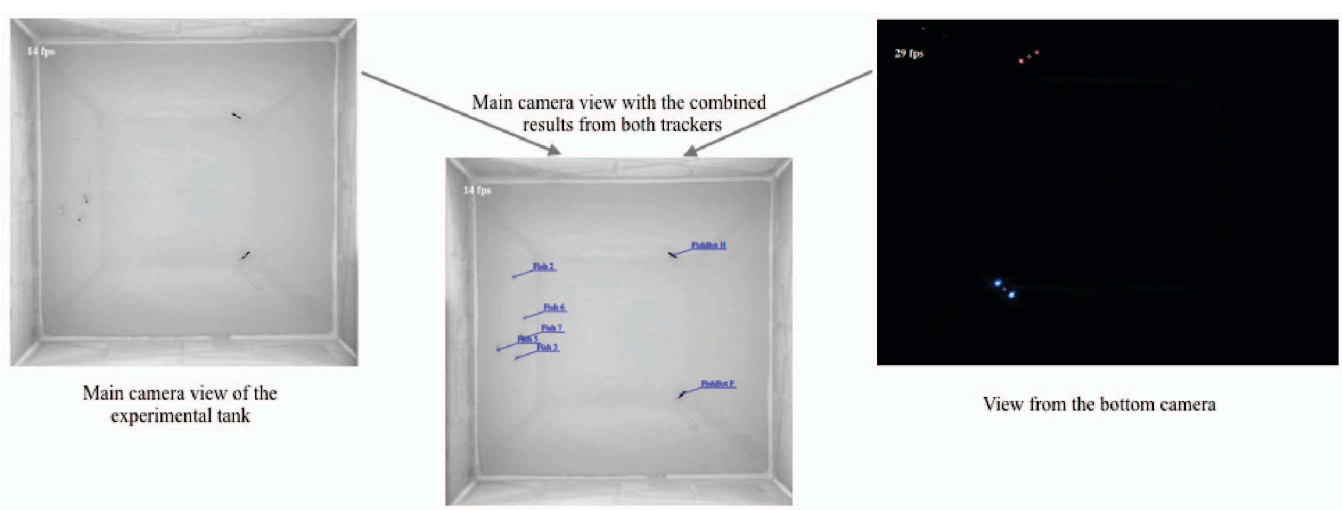

Fig. 6. The merge of the tracking results from (left) the camera above the setup that tracks all the agents, RiBots and fish, and (right) the camera below the setup that tracks only FishBots. The tracking results from both cameras are converted from pixels to millimetres thanks to the camera calibration routine and then merged together to detect robots among all agents.

a PID controller based on the difference of the current pose of the FishBot and the target pose. The gains of the PID (K_p, K_i and K_d) can also be parametrised, as they will depend on the linear speed of the robots or the type of arena that we are testing. The fish-like pattern is a finite state machine implemented at the level of the FishBot that follows a sequence of three states: orientation, acceleration and relaxation, that mimics the zebrafish locomotion underwater, while following a given target. The parameters of control modes and locomotion patterns can be modified via the user interface of CATS.

The "control map" mentioned above is a grey scale image file encoding the desired control mode and locomotion pattern for every part of the experimental arena. It's used if in some experiments we need the robot to execute different behaviour when in specific areas, for instance close to a border.

F. Zebrafish collective motion model

The model used in the new "fish-model" control mode mentioned in the previous section is based on zebrafish collective motion model proposed in [32]. In this model, the agents update their position vector X_i with a velocity vector V_i :

$$X_i(t + \delta t) = X_i(t) + V_i(t)\delta t \quad (1)$$

$$V_i(t + \delta t) = v_i(t + \delta t)\Theta_i(t + \delta t) \quad (2)$$

with v_i the linear speed of the i^{th} agent and Θ_i its orientation. The linear speed v_i of the agent is randomly drawn from the instantaneous speed distribution experimentally measured.

The orientation Θ_i is drawn from probability density function (PDF) computed as a mixture distribution of von Mises distributions centred on the stimuli perceived by the focal agent. In this study, we only took into account the influence of other agents and of the walls of the experimental arena. Thus, this custom PDF is composed by the weighted sum of (i) a PDF taking into account the effect of the walls and (ii) a PDF describing the response to other agents. The process is shown in Fig. 7.

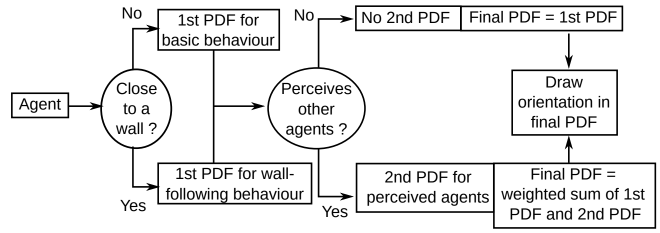

Fig. 7. Description of the different steps to compute the orientation Θ_i of a focal fish at each time step. The proximity to a wall is determined by comparing the distance of the agent to the closest wall with a threshold values d_w = 15cm.The PDF for basic behaviour is given by a von Mises distribution centred on 0 while the PDF for wall-following behaviour is given by a weighted sum of two von Mises distribution, each of them centred on one of the to possible direction along the wall. Other agents are perceived if they are present in the fov of the focal agent. The 2^{nd} PDF is then computed by weighting von Mises distributions centred on all perceived agents.

III. DEMONSTRATION

We have tested our framework for experiments involving mixed societies of robots and fish. First, we have succeeded in controlling multiple robots in closed-loop experiments for periods of up to 3 hours without any human interventions. The closed-loop control was performed in real-time, with robots adapting to the fish behaviours following for instance the model presented in Sec. II-F.

In order to demonstrate the functionalities of CATS software, we have created an arena to confine the mixed society of robots and animals that consisted of a circular arena of 580 mm diameter made of plexiglas (see Fig. 8). In this type of arena, the zebrafish are continuously moving, and the robot are able to follow the fish shoal when using the bio-mimetic "fish-model" described in Sec. II-F. We have used for this experiment a mixed society of three fish-robots and three zebrafish.

Figure 9 presents examples of individual trajectories from fish and robots in a setup with a circular arena. Robots can integrate into zebrafish groups, that tend to follow the walls of the arena, while mimicking their locomotion patterns. The fish are not afraid of the robotic lure, and we show the presence of closed-loop interactions: robots tend to follow the fish, and fish can follow a robot.

Figures 10 and 11 show the individual speed and inter-individual distance for each agent. In Fig. 10, we can observe that, in terms of mean and distributions, the robots are able to mimic the zebrafish in terms of speed. However, regarding the inter-individual distances, the fish are able to swim more close than the robot, as it can be observed in Fig. 11. The reasons for that is that the vision-based model do not guarantee that the target will always be generated by maintaining a fish-like shoaling distance between the agents. Also, sometimes, near the walls, the robots will have some difficulty to avoid the walls and mimicking the fish behavior that are moving in shoal along the walls. However, this is the first time that the shoaling capacities of a multi-agent platform is demonstrated to mimic the shoaling behavior of fish while being inserted into the animal group.

Our results are promising, and could be extended to more complex setups, with larger populations of fish and robots. A more thorough analysis of the dynamics of collective behaviours between fish and robots will be presented in a subsequent studies.

IV. CONCLUSION

We introduced a novel framework to perform long-duration experiments with a mixed society of several fish and robots. The presented CATS software is highly modular, flexible and efficient. It is able to track fish and robots in real-time, and to control several robots to exhibit reactive and complex behaviours. Thanks to the tracking of the FishBots from below, the shape of the lure or the type of enrichment inside the aquarium is not a constrain for the closed-loop

Fig. 8. The view of the experimental setup (described in Fig. 2 with a circular arena (diameter of 580 mm) made of plexiglas. We used three fish-robots and three zebrafish.

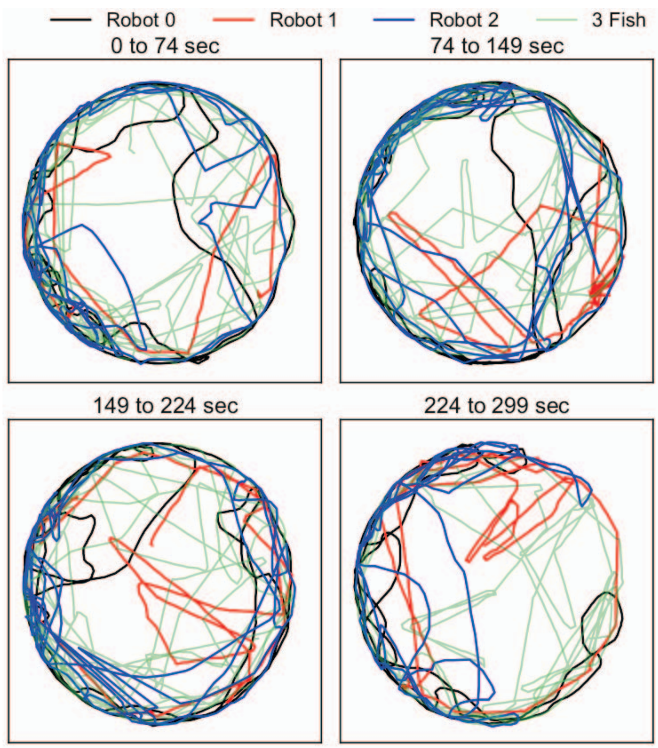

Fig. 9. Examples of individual trajectories from fish and robots in a circular arena. We used three fish-robots and three zebrafish. 300 seconds (4500 frames) are represented.

control of the robotic agents. Finally, the robots can be driven by a bio-mimetic model inspired from the zebrafish behaviour [32] to have closed-loop interactions between fish and robots.

Our framework has already been implemented into two universities for research involving mixed societies of animals and robots. (University Paris Diderot in France, and Ecole Polytechnique Fédérale de Lausanne in Switzerland). It was also implemented during the ARS-ELECTRONICA 2016 festival in Linz, Austria where the mixed society of fish and robot was presented to the public.

Additional research could be done to make robots more integrated with the fish group. Our methodology is designed to be generic, and could be potentially used to perform bio-hybrid experiments with other setups, involving other animal species.

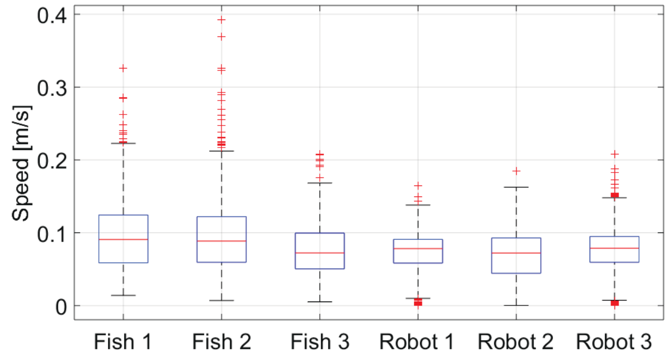

Fig. 10. Agents speed distributions in the mixed group experiment in a circular arena. 300 seconds (4500 frames) are represented.

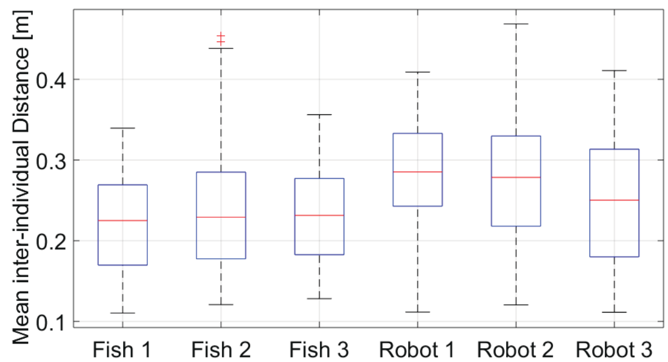

Fig. 11. Inter-individual distance distributions between agents in the mixed group experiment in an open circular arena. 300 seconds (4500 frames) are represented.

V. Acknowledgement

This work was supported by the EU-ICT project ASSISIbf, No. 601074. The information provided is the sole responsibility of the authors and does not reflect the European Commissions opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

References

[1] G. Sempo, S. Depickère, J. Amé, C. Detrain, J. Halloy, and J. Deneubourg, "Integration of an autonomous artificial agent in an insect society: experimental validation," in *From Animals to Animats 9*. Springer, 2006, pp. 703–712.

[2] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tâche, I. Said, V. Durier, S. Canonge, and J. Amé, "Social integration of robots into groups of cockroaches to control self-organized choices," *Science*, vol. 318, no. 5853, pp. 1155–1158, 2007.

[3] A. Gribovskiy, J. Halloy, J. Deneubourg, H. Bleuler, and F. Mondada, "Towards Mixed Societies of Chickens and Robots," in *IROS*, 2010.

[4] F. Zabala, P. Polidoro, A. Robie, K. Branson, P. Perona, and M. Dickinson, "A simple strategy for detecting moving objects during locomotion revealed by animal-robot interactions," *Current Biology*, vol. 22, no. 14, pp. 1344–1350, 2012.

[5] T. Landgraf, M. Oertel, D. Rhiel, and R. Rojas, "A biomimetic honeybee robot for the analysis of the honeybee dance communication system." in *IROS*, 2010, pp. 3097–3102.

[6] K. Griparic, T. Haus, D. Miklic, and S. Bogdan, "Combined actuator sensor unit for interaction with honeybees," in *Sensors Applications Symposium (SAS), 2015 IEEE*, April 2015, pp. 1–5.

[7] T. Landgraf, H. Nguyen, J. Schröer, A. Szengel, R. J. Clément, D. Bierbach, and J. Krause, "Blending in with the shoal: robotic fish swarms for investigating strategies of group formation in guppies," in *Biomimetic and Biohybrid Systems*. Springer, 2014, pp. 178–189.

[8] F. Bonnet, S. Binder, M. E. de Oliveria, J. Halloy, and F. Mondada, "A miniature mobile robot developed to be socially integrated with species of small fish," in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, Dec 2014, pp. 747–752.

[9] G. Polverino and M. Porfiri, "Zebrafish (danio rerio) behavioural response to bioinspired robotic fish and mosquitofish (gambusia affinis)," *Bioinspiration & biomimetics*, vol. 8, no. 4, p. 044001, 2013.

[10] S. Butail, G. Polverino, P. Phamduy, F. Del Sette, and M. Porfiri, "Influence of robotic shoal size, configuration, and activity on zebrafish behavior in a free-swimming environment," *Behavioural brain research*, vol. 275, pp. 269–280, 2014.

[11] ——, "Fish-robot interactions in a free-swimming environment: Effects of speed and configuration of robots on live fish," in *Proceedings of SPIE 9055, Bioinspiration, Biomimetics, and Bioreplication 2014*, 2014, pp. 90 550I–90 550I.

[12] D. T. Swain, I. D. Couzin, and N. E. Leonard, "Real-time feedback-controlled robotic fish for behavioral experiments with fish schools," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 150–163, 2012.

[13] Q. Shi, H. Ishii, Y. Sugahara, A. Takanishi, Q. Huang, and T. Fukuda, "Design and control of a biomimetic robotic rat for interaction with laboratory rats," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 4, pp. 1832–1842, 2015.

[14] T. Landgraf, D. Bierbach, H. Nguyen, N. Muggelberg, P. Romanczuk, and J. Krause, "Robofish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live trinidadian guppies," *Bioinspir Biomim.*, vol. 11, no. 1, 2016.

[15] D. Swain. (2011) Madtrac framework source documentation. [Online]. Available: http://poincare.princeton.edu/madtrac/docs/

[16] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, May 2009.

[17] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[18] Why ROS 2.0? [Online]. Available: http://design.ros2.org/articles/why_ros2.html

[19] F. Bonnet, P. Retornaz, J. Halloy, A. Gribovskiy, and F. Mondada, "Development of a mobile robot to study the collective behavior of zebrafish," *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 437–442, 2012.

[20] F. Bonnet, Y. Kato, J. Halloy, and F. Mondada, "Infiltrating the zebrafish swarm: design, implementation and experimental tests of a miniature robotic fish lure for fish–robot interaction studies," *Artificial Life and Robotics*, vol. 21, no. 3, pp. 239–246, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10015-016-0291-8

[21] F. Bonnet, N. Crot, D. Burnier, and F. Mondada, "Design methods for miniature underwater soft robots," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, June 2016, pp. 1365–1370.

[22] V. Braitenberg, *Vehicles: Experiments in synthetic psychology*. MIT press, 1986.

[23] Qt. cross-platform software development for embedded & desktop. [Online]. Available: https://www.qt.io

[24] Aravis - a vision library for genicam based cameras. [Online]. Available: https://github.com/AravisProject/aravis

[25] gstreamer - open source multimedia framework. [Online]. Available: https://gstreamer.freedesktop.org

[26] A fast single-producer, single-consumer lock-free queue for c++. [Online]. Available: https://github.com/cameron314/readerwriterqueue

[27] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

[28] J. Shi and C. Tomasi, "Good features to track," *Computer Vision and Pattern Recognition, Proceedings CVPR*, 1994.

[29] J. Serra and P. Soille, Eds., *Mathematical Morphology and Its Applications to Image Processing*, 1994.

[30] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[31] R. Y. Tsai, "Radiometry," L. B. Wolff, S. A. Shafer, and G. Healey, Eds., 1992, ch. A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, pp. 221–244.

[32] B. Collignon, A. Séguret, and J. Halloy, "A stochastic vision-based model inspired by zebrafish collective behaviour in heterogeneous environments," *Royal Society Open Science*, vol. 3, no. 1, p. 150473, 2016.