

PiRat: An autonomous framework for studying social behaviour in rats and robots

Robotics lab: Scott Heath¹, Carlos Andres Ramirez-Brinez¹, Joshua Arnold¹, Ola Olsson¹,
Jonathon Taufatofua¹, Pauline Pounds¹, Janet Wiles¹

Rat lab: Eric Leonardis², Emanuel Gygi², Estelita Leija², Laleh K. Quinn², Andrea A. Chiba²

Abstract—The use of robots, as a social stimulus, provides several advantages over using another animal. In particular, for rat-robot studies, robots can produce social behaviour that is reproducible across trials. In the current work, we outline a framework for rat-robot interaction studies, that consists of a novel rat-sized robot (PiRat), models of robotic behavior, and a position tracking system for both robot and rat. We present the design of the framework, including constraints on autonomy, latency, and control. We pilot tested our framework by individually running the robot rat with eight different rats, first through a habituation stage, and then with PiRat performing two different types of behaviour — avoiding and frequently approaching. We evaluate the performance of the framework on latency and autonomy, and on the ability to influence the behaviour of individual rats. We find that the framework performs well on its constraints, engages some of the rats (according to the number of meetings), and features a control scheme that produces reproducible behaviour in rats. These features represent a first demonstration of a closed-loop rat-robot framework.

I. INTRODUCTION

There is an increasing use of robots in social interaction studies on rats (e.g., [1], [2]) as well as on other animals (e.g., fish [3], penguins [4]) [5]. Evoking and measuring neural and behavioural responses to a carefully controlled social stimulus is a powerful tool for studying and understanding the rodent brain. There are several advantages to using embodied robots in these contexts: i) running online models of behaviour, enables the implementation of what may be perceived as a social “other” while their behaviours can be well described over time, ii) interaction with well-described robotic partners can act as an important comparison case for the interaction between two mammals, iii) a social robot allows for the experimental manipulation of physical and sensory properties/characteristics, to assess the possible critical features that drive social behaviour, and iv) robots allow experimenters to create reproducible behaviours or adaptive models, allowing investigation of the importance of adaptability and responsiveness for social interaction.

When creating a robot as a social stimulus for a rat, it is not feasible to implement the entire neural architecture of a rat; so part of the challenge is identifying which social

This work was funded by [NSF BRAIN EAGER 1451221; NIMH R01 5R01MH110514-02; AOARD FA2386-16-1-4026]

¹Engineers are with the School of ITEE at the University of Queensland, St Lucia, Queensland, Australia scott.heath@uq.edu.au

²Neuroscientists are with the Department of Cognitive Science, University of California, San Diego, La Jolla, San Diego, CA

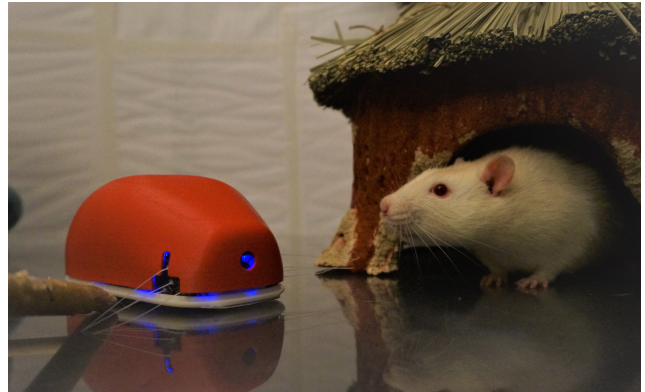


Fig. 1. A rat interacts with a stationary PiRat. Rats were willing to approach and inspect PiRat. PiRat’s whiskers are new sensors that are currently under development, but are not used in the current study.

features are important for a robot to implement and exhibit to the rat. Previous studies have approached this using wizard-of-oz (WoZ) control of the robot [1], or using autonomous robots [2]. The benefit of using a WoZ style of control is that the experimenter can create the behaviour for the robot using all of the sensory and cognitive apparatus of the human. However, WoZ requires expertise and different experimenters will not be able to reproduce exactly the same behaviour. The benefits of autonomy are that the control of a robot is consistent with their algorithms. The difficulties with autonomy are in the implementation: sensing and representing the environment with enough detail to be able to act socially. In practice, all studies with autonomous agents have some degree of human involvement.

The aim of the PiRat project is to develop and iteratively improve a set of rat-sized robots and related software and hardware in order to study rat social interactions. The key challenge in achieving this goal is creating the capacity for the robot to adapt to the rat’s behaviour. The ability to adapt behaviour to the individual differences, encountered in the social world, is essential for successful social interactions.

In this paper, we describe the development and evaluation of a rat-sized robot and tracking framework for closed-loop, rat-robot interaction studies. The components of this system are described in detail. The design of a new robot platform is described, along with a new tracking system that is capable of determining the location and orientation of the

robot, and a location estimate for a real rat. Together, the robot and tracking system are used to create a closed-loop controller, enabling the robot to perform a series of simple behaviours that rely on knowing the location of both robot and rat. The new system is pilot tested by investigating short interactions between the robot and eight individual rats. A set of interaction summary metrics are proposed, and collected position data from the study is analysed using these metrics.

The paper is organised as follows: Section II presents related work on robot-rodent studies and similar tracking systems, Section III describes the design and control of the robot and framework for interaction studies, Section IV describes the pilot rat-robot interaction study performed with the robot and eight rats, Section V discusses the implications of this contribution, and finally we draw conclusions about the usability of this system for rat-robot interaction studies.

II. RELATED WORK

Autonomous frameworks developed for rat social stimulus interactions typically feature a robot platform or software agent, sensors that can give feedback on the status of the rat, and additional computers for computation. Previous studies have tended to focus on one of these aspects. The Waseda rat studies are the state-of-the-art for rat-robot design [6], [7]. The Waseda group has designed a series of robots that can emulate specific rat social cues, such as rearing and grooming. Their limbed robot rats are able to perform unique rat-like behaviours, and they demonstrate their robots performing simple interactions with real rats. Notably, the Waseda rats illustrate the need for fast movements as later models use a combination of wheels and legs in order to perform both rat-like behaviours and locomote quickly. However, the Waseda rats are being used for the purpose of eliciting negative emotional states (fear, stress, and depression) or to act as a punishing stimulus in laboratory rats, as opposed to examining behaviours that result in periodic social interactions, or even pro-social behaviours. Other studies have focused on rat-robot interactions and used off-the-shelf robot platforms instead of building their own [8], [2]. These studies differ in the type of interaction, robot platforms and sensory apparatus. Ortiz et al. use an e-Puck [9] with simple behaviours to evaluate social responses to the robot from rats [2]. The robot they use is completely autonomous; however, the sensors on the robot (infra-red sensors) limit both the robot's representation of its environment, and the data that can be collected from the study. Gianelli et al. use a Sphero robot to teach a rat how to follow specific paths [8]. They have an overhead tracking system based on an RGB camera that is capable of tracking both rat and robot. These studies feature real-time autonomous robots that are capable of interacting with rats. Furthermore, the framework is applicable to a wide range of spatial navigation behavioural studies. The main limitations are around the behaviours of the robot - the robot moves along a set of waypoints, but does not seem to be able to use information about the real rat in planning. Additionally, the methodology of providing food rewards from the robot to the rat is not

applicable to studies looking at social interaction, as the rat may subsequently treat the robot as a mobile food dispenser.

In our previous collaborative work, we investigated rat-robot interactions using our previous generation rat-sized robot, the iRat [10]. We were able to pilot-test the iRat through observational studies of rat-robot interactions [1], and more recently, we were able to demonstrate that rats will release a trapped iRat robot as part of our studies examining pro-social behaviours [11]. Whereas the iRat was able to evoke interesting behaviours, its autonomy and sensors were limited.

A. Tracking systems

Several commercial and open-source tracking systems are available (e.g., [12], [13]); however, we found that none of them fit our set of constraints. Our constraints included: i) running in real-time, ii) working in light and dark conditions, iii) tracking both rat and robot, iv) non-invasive, and v) open-source so that the software in the PiRat framework can be freely distributed and extended to new behaviour paradigms. Many of the available tracking systems were designed for offline analysis only, so were unsuitable for use with a robot. The only real-time options with potential were closed-source, commercial solutions, where the proprietary and opaque nature of the software make them unsuitable for the PiRat framework.

III. DESIGN AND CONTROL OF A ROBOT RAT

The PiRat system is designed to address the limitations of previous studies with regards to autonomy, and sensors, and to give more control of the study to the experimenters. The PiRat system is formed from several components including the robot platform, a Kinect v2 for tracking, a router for connecting components together and a laptop (see Figure 2). Multiple software components run on the laptop, including the tracking system, the behaviour manager and MATLAB running a ROS server. Each of these components is outlined in the following sections. For a video overview of the framework and experimental set up, see <https://www.youtube.com/watch?v=uzqJsvfEnr4>.

A. PiRat — the second generation iRat

PiRat is intended to be a smaller and faster version of our previous iRat [10]. Initial specifications were collected through extensive dialog between engineers and neuroscientists (see Table I).

PiRat consists of a shaped plastic shell that houses two *gimbal motors* — high-speed, brushless DC motors — and circuit boards that are required to drive the motors and communicate over WiFi. The shell is constructed from 2 pieces, both of which are 3D printed using a Form2 3D printer (see Figure 3a). The two pieces of the shell are held together using magnets, so that the top half can be quickly removed (necessary for replacing PiRat's batteries and shell cleaning, see Figure 3b). The motors are mounted on a separate piece of plastic (see Figure 3c), which allows both motors to be easily removed and cleaned. During interactions

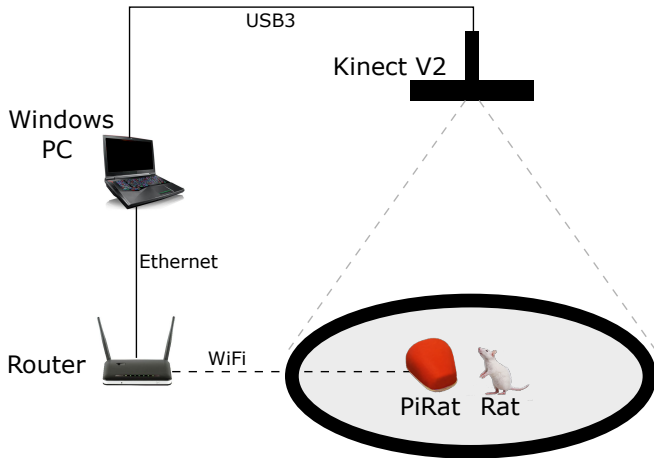


Fig. 2. The architecture of the setup. The Kinect V2 that is used for tracking is connected to a Windows PC that runs tracking, behaviour manager, and a ROS master. Velocity commands are sent to PiRat via a router.

TABLE I
INITIAL AND IMPLEMENTED PiRAT REQUIREMENTS

Requirement	Value	Actual
Width	< 0.10 m	0.0875 m
Length	< 0.15 m	0.123 m
Height	< 0.08 m	0.055 m
Max. Speed	1m/s	1.1m/s
Max. Angular Velocity	π rad/s	4.7rad/s
Weight	0.25 kg	0.24 kg
Audio frequencies	< 10 dB over 25 kHz	✓
GUI	Allow selecting behaviours	✓
	Use XBOX controller	✓
Autonomy	Behaviours autonomous	✓
	Needs to know where rat is	✓
Cleanliness	Shell and motors cleanable	✓

with rats, robots can become dirty, so having the shell and motors separately cleanable is a key design consideration of PiRat.

PiRat electronics consist of three circuit boards stacked one on top of the other having an identical footprint to that of a Raspberry Pi Zero (Pi Zero). The top board in the stack is the Pi Zero (running Raspbian), allowing PiRat to connect to Wi-Fi networks through its inbuilt wireless card. The Pi Zero is connected to the distribution board through a novel USB connector that slots in at a right-angle to the other two boards (see Figures 3d and 3e). The second board (the distribution board) contains an STM32 F042 microcontroller that handles the USB connection to the Pi Zero, PWM connections to magnetic wheel encoders and a USART connection to the third circuit board. The third circuit board (the driver board) is a custom-made two gimbal motor driver based on Martinez

Gimbal board¹) that generates the control signals for PiRat's gimbal motors.

Software on PiRat is minimal. The Pi Zero uses Robot Operating System (ROS) to communicate with a tracking system and behaviour manager (see Figure 3f for all hardware). Translational and angular velocities are forwarded from the Pi Zero to the gimbal motors. PiRat weighs 0.24 kg, has a top speed of 1.1m/s and a top angular velocity of 4.7m/s.

The PiRat also includes Pi Zero camera and two new whisker sensors (that are still under development), neither of which are used in the current study.

B. Tracking system

Behavioural experiments using rats are commonly performed in both light and dark conditions, making it important for the tracking system to be robust in both cases. Dark environments makes color- or contrast-based tracking schemes difficult to apply, especially as rats come in both light and dark varieties. Because of this we opted for a depth-based tracking system using the Kinect V2 sensor, which produces high-quality point clouds. The Kinect V2 is based on time-of-flight and uses its own infrared light source, making it relatively insensitive to ambient lighting conditions. For the rigid PiRat we used a tracking method based on Iterative Closest Points (ICP) [14], using a template point-set sampled from the CAD model of PiRat. This provides tracking of both location and orientation with sufficient precision to enable closed-loop control. However, this approach is only suited for rigid objects, so will not work for real rats. We therefore implemented a simpler scheme, tracking only the rough position of the rat. This enabled the system to use the position of the rat (but not its posture) to generate adaptive behaviours. The implemented method optimizes a hemispherical template using a cost function that takes into account the distance and orientation of sample points from the template using gradient descent. The resulting system autonomously tracks PiRat and rat during studies. The system does not support detection, only tracking, and therefore user interaction is required to place the template on the target at the start of an experiment, and again if tracking is lost during the experiment by clicking on the tracker UI (see Figure 4a). The tracking system is implemented in C++ for speed, and poses are sent using sockets to the PiRat behaviour manager, which is typically running on the same Windows machine. Key benefits of our tracking system are, real-time performance, that the rat does not have to be instrumented, and that the tracking can be corrected on-line.

C. PiRat behaviour manager

The behaviour manager is a Python program that controls PiRat through a set of simple behaviours. Examples of behaviours are: approaching, following, retreating, exploring, avoiding, and collision-avoiding. The manager maintains a simple model of the environment (the tracked positions

¹See <https://sourceforge.net/projects/brushless-gimbal-brugi/>

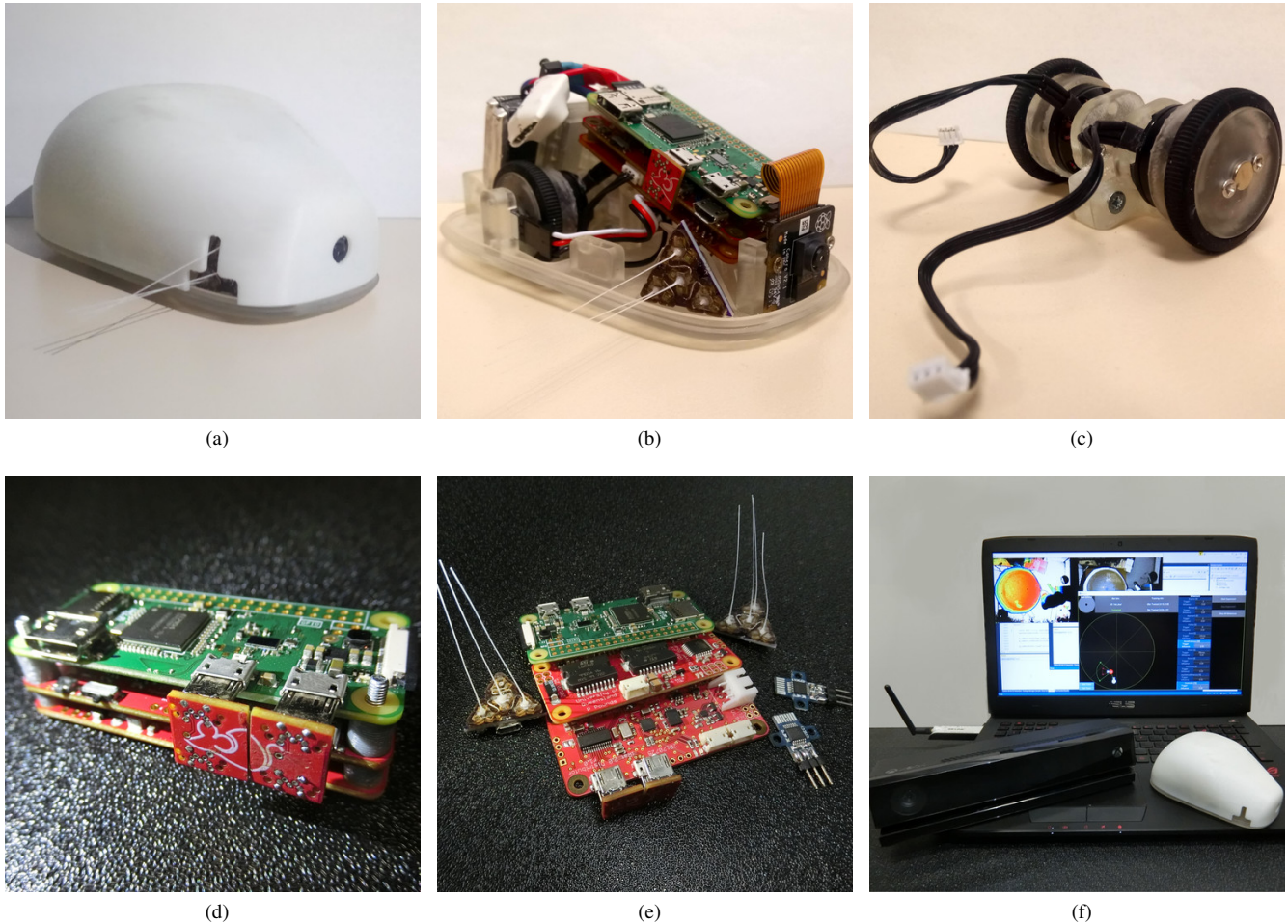


Fig. 3. Different components of the PiRat hardware. a) the PiRat, b) the PiRat without shell, c) the PiRat's removable wheels, d) the PiRat PCB stack with the Pi Zero on top, e) the PiRat PCB stack disassembled, and f) the external hardware components of the PiRat.

of PiRat and rat), which are the only inputs used by the behaviours (PiRat sensors are not used in this study). Only the approaching, avoiding, and collision-avoiding behaviours are used in this study.

Approaching: PiRat calculates the distance between itself and the rat, and then moves towards the rat until the distance is half of the initial value.

Avoiding: PiRat calculates the direction of the rat from the centre of the arena, then moves towards the diametrically opposite point on the boundary of the arena.

Collision-avoiding: When PiRat comes within close proximity of the rat or arena edge, forward velocity is slowed and angular velocity increased in proportion to the direction and distance away from the obstacles.

The manager additionally implements a random selection of the behaviours, such that behaviours can be run for a random amount of time, or until a natural completion condition before switching to another behaviour. Multiple behaviours can simultaneously control the iRat's actuators through a time dependent influence that is used to mix the linear and angular velocities from each active behaviour together. PiRat's linear velocity, v , and angular velocity, ω ,

are calculated as follows:

$$v = \sum_i \alpha_i(t) \cdot v_i$$

$$\omega = \sum_i \alpha_i(t) \cdot \omega_i$$

across all i behaviours. In these studies, it is typical for the approaching or avoiding behaviour to be active at the same time as the collision-avoiding behaviour. In this case PiRat will give priority to avoiding obstacles (the real rat, or the sides of the arena) while attempting to approach or avoid the real rat. Closely integrated with the behaviour manager, in the same Python program, is a PiRat GUI. The GUI uses the UI framework Kivy to display the current state of PiRat and real rat (see Figure 4b). Direct control and triggering of behaviours of PiRat is also implemented through an XBOX controller, or through the keyboard.

D. Study environment

The PiRat and tracking system are designed for use in a circular environment, with a single PiRat and a single rat (at this stage). The arena has a diameter of 1220 mm with

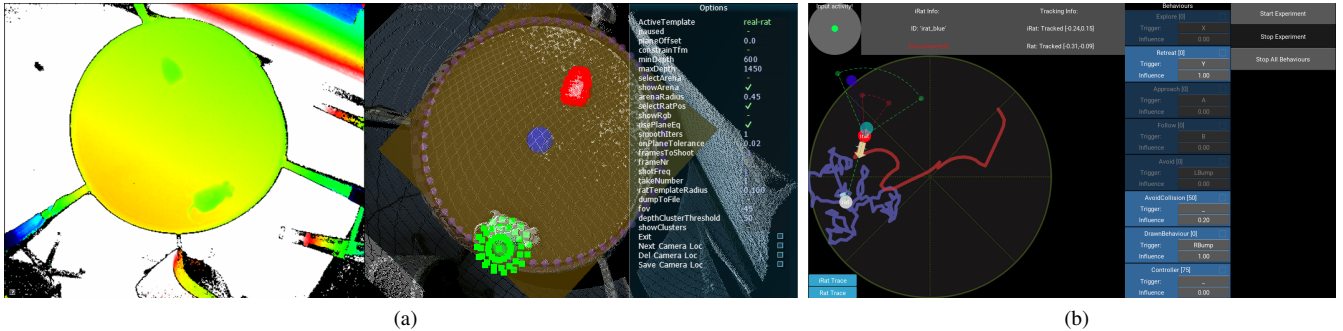


Fig. 4. PiRat GUIs. a) the GUI used for tracking, b) the GUI used for managing PiRat behaviours.

TABLE II
LATENCIES OF DIFFERENT PARTS OF THE FRAMEWORK

Latency Source	Latency (ms)
Kinect V2 input latency	≈ 80 (≈ 100 screen to screen)
Tracking processing latency	≈ 24
GUI + iRat command latency	≈ 22
Total system latency	≈ 126

a wall around the outside of height 305 mm. Other objects can be placed in the environment (provided the shape is not too similar to that of PiRat).

E. Ethics

All animal experiments and maintenance procedures were performed in accordance with NIH and local IACUC guidelines at UCSD (an AAALAC accredited university).

F. Latency

An important consideration in the design of an autonomous interactive robot is latency to respond. For humans, latency has been demonstrated to affect perceived causality [15]. In contrast, for rats, the effects of interactive latency deficits in embodied robotic systems have yet to be described. It is important to characterise the latency of such a system for comparison with other rat-robot interaction frameworks, and to increase our understanding of how these latency deficits affect rat-robot interactions.

We characterised the latency of our system by sending oscillating velocities to PiRat and logging the velocity and position reported by the tracking system at the GUI. We then fit sine waves to both the sent velocities, and to the position reported by the tracker. Calculating the difference in phase offsets gave an estimate of the latency of the overall system as 126 ms. Assuming that the Kinect v2 has an input latency of $\approx 80^2$, and the tracking latency could be directly calculated in software as 23.8 ms, the GUI + PiRat latencies were estimated ≈ 22.2 ms.

²We measured screen-to-screen round-trip latency of the Kinect v2 to be ≈ 100 ms. We estimate that this comprises ≈ 80 ms for input from the Kinect, and another ≈ 20 ms for the image to appear on the screen.

G. Summary metrics

Metrics are required in order to compare across or within interactions between a rat and the PiRat. Ortiz et al. proposed a series of metrics for their rat studies, and demonstrate that the occurrences of individual behaviours are more significant across their conditions than the time spent performing behaviours [2]. We propose three simple metrics for an interaction. These metrics are based on values that are intuitively linked to close interaction, and are also calculable from the data collected by our system. The metrics are: i) mean and standard deviation of the distance between the two interactants, ii) number of meetings (where the distance between rat and PiRat drops below a given threshold), and iii) average component of speed in the direction of PiRat.

IV. PILOT STUDY

We performed a pilot study to test the framework on a simple experimental paradigm.

A. Aims

The aim of the pilot test was to evaluate the different responses of the rat to PiRat performing 2 different composite behaviours, i) PiRat always avoids the rat (avoid behaviour previously described), and ii) PiRat changes between avoiding the rat and approaching half-way to where the rat currently is (a combination of avoid and approach behaviours).

B. Setup

The study consisted of rat-robot interaction trials with 8 rats (6 Sprague-Dawley and 2 Brown Norway rats) over 2 days. All studies were run with the room lights on. The procedure was as follows:

Day 1: Habituation

- 1) A rat is placed into the arena alone for 1 minute.
- 2) The rat is then removed from the arena.
- 3) The rat is placed into the arena with a stationary iRat for 1 minute.

Day 2: Trials

- 1) A rat is placed into the arena with the PiRat running circles of the arena (the Introduction) for 2 minutes.
- 2) The rat is removed from the arena.

- 3) The rat is placed in the arena with the PiRat running either the Avoid or Approach condition (counterbalanced across subjects) for 2 minutes.
- 4) The rat is removed from the arena.
- 5) The rat is placed in the arena with the PiRat running the second condition (Avoid or Approach) for 2 minutes.

Overhead tracking data was logged (complete RGB-D frames were stored at 30 Hz) for Introduction, Approach and Avoid sessions. Results were analysed through visualisation, and by applying our proposed summary metrics.

C. Results

The RGB-D frames were post-processed using the tracking software offline. The positions of PiRat and rat were recorded for each frame. The positions taken from tracking the rat and PiRat show the different trajectories of the rat and PiRat in each condition (Figure 5). Distances between the rat and PiRat were calculated for each study.

Metrics for the rats' responses to the robot yielded similar values across the approach and avoid behaviours, but were quite different for the Introduction when compared to the other behaviours (see Table III). For the mean distance between rat and PiRat, the Introduction shows the lowest number across all the rats, while the approach behaviour is lower than the avoid behaviour for 7 of the 8 rats. This difference between avoid and approach could be explained by the behaviour of PiRat alone. For the 1 rat where the metric is lower for the avoid behaviour, the PiRat started an approach from very close to the rat and spent a significant period following the rat around before reaching half the distance.

The Introduction has the highest number of meetings for all rats, and lowest mean distance between for 7 of the 8 rats, but the component velocities in the direction of PiRat vary. This can be explained by the rat's behaviour where in many of the Introduction trials, the rat would run ahead of PiRat, and then wait for PiRat to catch up (see component velocities in Figure 5).

One rat, S2, has the highest number of meetings, and shows differences from the other rats across all the phases. In particular, the velocity components in the direction of PiRat show that this rat was more active in creating meetings. In observations of the study, S2 often moved in front of PiRat in all the phases, temporarily blocking the robot from proceeding. Based on this observation, further exploration of the way in which the dominant or subordinate behavioral phenotype of the rat impacts its interactions with robots should prove interesting.

D. Evaluation of the tracking system

The tracking system was rerun using the collected depth and RGB-D data from the recordings during PiRat-Rat interactions. The number of times the system had to be readjusted due to losing tracking of either the rat or PiRat, gives an indication of autonomy of the system. This metric was chosen in preference to the number of errors, which is not a reliable measure as it depends on the speed of human intervention. In 13/24 (54%) offline trials, the tracking

TABLE III
INDIVIDUAL RAT-ROBOT INTERACTIONS

Rat ID	Study phases								
	Intro			Approach			Avoid		
	μ	#	d	μ	#	d	μ	#	d
BN2	0.58	8	48.22	0.51	5	21.02	1.07	0	62.16
BN3	0.57	8	19.58	0.87	0	20.64	1.09	0	19.20
EG5	0.58	10	36.76	0.77	0	42.97	1.02	1	38.01
EG6	0.49	12	28.53	0.82	0	33.90	1.10	0	37.78
EG8	0.53	12	31.61	0.84	0	31.27	1.09	0	18.46
R27	0.62	11	43.48	0.85	0	31.29	1.07	0	37.97
S1	0.62	11	44.39	0.85	0	13.63	1.09	0	14.80
S2	0.54	17	58.42	0.78	2	40.28	0.98	2	97.98

μ = the mean distance between rats (m)

= number of meetings (distance dropped below 0.2 m)

d = the sum of the positive components of rat velocity in direction of PiRat (m)

system was completely autonomous and required no human intervention. For the remaining trials, the reasons for failure were rats climbing onto the edges of the arena, or the tracking system confusing PiRat with the rat after a meeting. Tracking of PiRat was very reliable, PiRat was only lost twice throughout the offline tests, giving a 91% success rate. When tracking of the rat was lost, human intervention (via a single click) was used to restore the tracked location of the rat, and tracking resumed normally from that point. Most tracking difficulties were encountered during the Introduction, as the larger number of meetings caused the tracker to confuse the rat for PiRat more frequently.

V. DISCUSSION AND CONCLUSIONS

In this study, we aimed to create a rat animat and closed-loop control system that was capable of adapting its behaviour to the state of a rat. Results showed that the rats took different trajectories according to the different behaviours of the robot. This result is an important benchmark in enabling a robot to autonomously interact with a rat, demonstrating the robot's relevance to the rat's behaviour. Metrics for close interaction suggest that the rat attended to the robot in several of the cases, with the highest quality interactions occurring in the Introduction behaviour.

We divided our behaviours into two different conditions: approaching, and avoiding. After initial tests, an Introduction phase was added before initiating the other behaviours, to facilitate the rats' comfort around PiRat. We found that the difference between how rats responded to these two behaviours was minimal; however, differences between these behaviours and the Introduction phase were notable, with all the rats meeting the iRats more times in this phase than the other two phases. This may be because the Introduction featured a predictable motion in a space that the rat would usually occupy (PiRat always following the walls), whereas the other behaviours depended on the rat's motion.

We proposed metrics that our system could calculate easily to summarise the quality of the interactions in the study, and calculated these metrics for each rat interaction. These are useful for categorising behaviour and identifying interesting individual rats. Calculation of these metrics suggest that

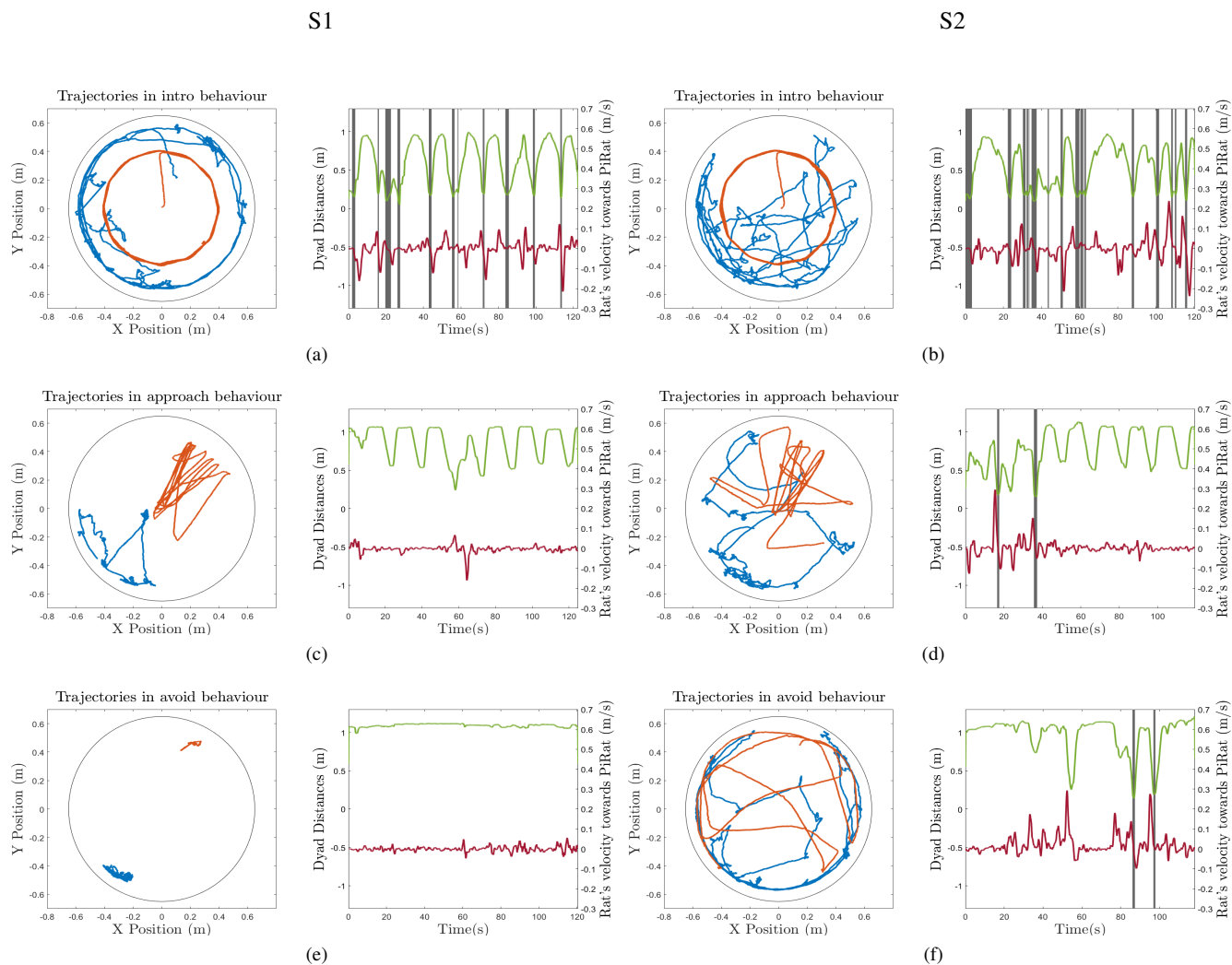


Fig. 5. Trajectories of, and distances between the rat and PiRat. The circular plots show the trajectories during the experiment between rat (in blue) and PiRat (in orange). The rectangular top plots show the distances between the rat and PiRat over each session. The bottom rectangular plot represents the component of the rat's velocity in the direction of PiRat. The black vertical bars in these plots show when an encounter occurs (the distance between rat and PiRat drops below 0.2m). a), c) and e) show the rat that had the least encounters (S1) across the Introduction, PiRat's approaching behaviour and PiRat's avoiding behaviour respectively. b), d) and f) show the rat that had the most encounters (S2) across the Introduction, approaching behaviour and avoiding behaviour.

further work on PiRat's behaviours is necessary for higher quality interactions. The first two metrics (mean, standard deviation, and number of meetings) are also influenced by the behaviour of PiRat, while the third metric (average rat velocity in direction of PiRat) better captures the rat's behaviours, but can not capture sequences where the rat deliberately waits for the PiRat to approach.

The framework presented in this paper was designed to facilitate rat-robot interaction studies through creation of an autonomous PiRat, by having the rat's position as a behavioural input to PiRat, by providing low-latency control of PiRat, and by making the system extensible to new behaviours. The framework was semi-autonomous, as a person was still required to correct tracking failures. However, the tracking for both rat and PiRat were robust, and the operation of the system required only occasional human intervention, in

contrast with the constant human input required in previous iRat studies. The increased autonomy is what enables PiRat to ensure reproducible experimental conditions. The human intervention has only a minor effect on reproducibility, as the human is not controlling any of the behaviours, instead intervention is limited to correcting the world-view of the framework.

Allowing the rat's position to be used as an input enables behaviours that are specific to each individual rat. While the rats in this study had more meetings with PiRat in the habituation phase, we hypothesise that a combination of further habituation, and changes that initially enable PiRat's behaviour to approach less directly, will positively impact the number of encounters that the rat and robot have. These changes will allow the rats time to accommodate to PiRat, reducing aspects of fear or perceived threat.

The framework's extensibility allowed us to rapidly develop new behaviours. New behaviours can be implemented in Python as a function that takes the robot's and rat's positions, and outputs translational and rotational velocities for PiRat. While developing new behaviours is quick, testing the behaviours with rats to gauge their effectiveness takes longer.

A. Rat-robot interaction studies

Our study has several implications for other rat-robot studies. Behaviour of the robot is a fundamentally important consideration, and we found that in our pilot, predictable behaviour led to more meetings initially. For longer interactions, further studies are required.

The degree of autonomy of the framework is also important. Frameworks that rely on WoZ control schemes, such as our previous work [10]), are susceptible to the stimulus not being reproducible. Conversely, completely reproducible frameworks that provide static behaviours, independent of the rat, cannot capture the individual interaction differences between rats. Our framework fits between these two cases, where PiRat's behaviour is algorithmically dependent on the rat's behaviour. This means that if the rat behaves identically in multiple sessions, then PiRat will also behave identically in these sessions.

For studies that use tracking as part of their behaviour, the latency from sensors to robot behaviours is important. In contrast, for studies that use tracking for post-processing, the framerate of the sensor is the most important factor. It is important to note that for many sensors, latency can be much slower than framerate, e.g., the Kinect v2 has a latency of 100 ms from sensor to screen, but is able to provide a new image to a screen every 33.33 ms. Our framework has to consider latency so that the real-time response of PiRat happens as close as possible to any change in the rat's position, and these initial estimates suggest that the system maintains a fairly low total systems latency within the real-time loop.

Our framework is flexible enough that it would be capable of running identical studies to those of both Ortiz et al and Gianelli et al with minor changes. While Ortiz et al. use sensors local to the robot, the same behaviours could be implemented in relation to the rat's and PiRat's absolute position. For the study of Gianelli et al. PiRat is already capable of using waypoints (as used for the habituation behaviour) and the only required addition would be the food dispenser. The advantage of our setup is that in both cases, the tracking of the rat is mostly-autonomous and non-invasive.

B. Limitations and future work

While we have demonstrated that our framework can be used to autonomously study rat-robot interactions, the system can further be improved. The input required from the experimenter can be further reduced (as we have already reduced human input compared with our previous studies [10]). We can extend the current tracking setup to richer environments and additional dynamic objects. Tracking does

not yet capture the facing direction of the rat, and the head and body direction of the rat is an important social input. Finally, the behaviours implemented on PiRat are still simple, but as the platform is extensible, we intend to extend the framework with further behaviours, model-driven approaches to social interaction, and the addition of PiRat's sensors as inputs to the framework. We intend for this new robot platform to allow us to continue exploring behavioural and neural responses of rats to a robot, allowing a systematic approach to the study of rat social behavior.

REFERENCES

- [1] J. Wiles, S. Heath, D. Ball, L. Quinn, and A. Chiba, "Rat meets iRat," in *Proceedings of the International Conference on Development, Learning and Epigenetic Robotics*. IEEE, 2012, pp. 1–2.
- [2] R. d. A. Ortiz, C. M. Contreras, A. G. Gutiérrez-Garcia, and M. F. M. González, "Social interaction test between a rat and a robot: A pilot study," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 4, 2016.
- [3] L. Cazenille, B. Collignon, Y. Chemtob, F. Bonnet, A. Gribovskiy, F. Mondada, N. Bredeche, and J. Halloy, "How mimetic should a robotic fish be to socially integrate into zebrafish groups?" *Bioinspiration & Biomimetics*, vol. 13, no. 2, p. 025001, 2018.
- [4] Y. Le Maho, J. D. Whittington, N. Hanuise, L. Pereira, M. Boureau, M. Brucker, N. Chatelain, J. Courtecuisse, F. Crenner, and B. Friess, "Rovers minimize human disturbance in research on wild animals," *Nature Methods*, vol. 11, no. 12, p. 1242, 2014.
- [5] A. Frohnwieser, J. C. Murray, T. W. Pike, and A. Wilkinson, "Using robots to understand animal cognition," *Journal of the Experimental Analysis of Behavior*, vol. 105, no. 1, pp. 14–22, 2016.
- [6] Q. Shi, H. Ishii, S. Miyagishima, S. Konno, S. Fumino, A. Takanishi, S. Okabayashi, N. Iida, and H. Kimura, "Development of a hybrid wheel-legged mobile robot WR-3 designed for the behavior analysis of rats," *Advanced Robotics*, vol. 25, no. 18, pp. 2255–2272, 2011.
- [7] Q. Shi, H. Ishii, K. Tanaka, Y. Sugahara, A. Takanishi, S. Okabayashi, Q. Huang, and T. Fukuda, "Behavior modulation of rats to a robotic rat in multi-rat interaction," *Bioinspiration & Biomimetics*, vol. 10, no. 5, p. 056011, 2015.
- [8] S. Gianelli, B. Harland, and J.-M. Fellous, "A new rat-compatible robotic framework for spatial navigation behavioral experiments," *Journal of Neuroscience Methods*, vol. 294, pp. 40 – 50, 2018.
- [9] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klapotcz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the Conference on Autonomous Robot Systems and Competitions*, vol. 1. Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.
- [10] D. Ball, S. Heath, G. Wyeth, and J. Wiles, "iRat: Intelligent rat animat technology," in *Proceedings of the Australasian Conference on Robotics and Automation*, 2010, pp. 1–3.
- [11] L. Quinn, L. P. Schuster, M. Aguilar-Rivera, J. Arnold, D. Ball, E. Gygi, S. Heath, J. Holt, D. J. Lee, J. Taufatofua, J. Wiles, and A. A. Chiba, "When rats rescue robots," *Animal Behaviour and Cognition*, Submitted 2018, In Press.
- [12] P. Aguiar, L. Mendonça, and V. Galhardo, "OpenControl: a free opensource software for video tracking and automated control of behavioral mazes," *Journal of neuroscience methods*, vol. 166, no. 1, pp. 66–72, 2007.
- [13] L. P. Noldus, A. J. Spink, and R. A. Tegelenbosch, "EthoVision: a versatile video tracking system for automation of behavioral experiments," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 398–414, 2001.
- [14] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [15] J. Nielsen, *Usability engineering*. Elsevier, 1994.